Data Mining in Tree-Based Models and Large-Scale Contingency Tables

A Thesis
Presented to
The Academic Faculty

by

Seoung Bum Kim

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Industrial and Systems Engineering
Georgia Institute of Technology
December 2004

Data Mining in Tree-Based Models and Large-Scale Contingency Tables

Approved by:

Dr. Kwok-Leung Tsui, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. David Goldsman
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Xiaoming Huo
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Wenke Lee
College of Computing
Georgia Institute of Technology

Dr. Joseph Wu
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: December 2004

To my grandmother, parents, and wife.

ACKNOWLEDGEMENTS

I would like to thank a number of individuals for their assistance and support in helping make this dissertation possible. Foremost, I would like to express my sincerest gratitude to my advisor, **Kwok Tsui**, for sharing his insight and knowledge in the field of statistics and for his wonderful guidance during my whole program of study. His enthusiasm and encouragement made the entire Ph.D. study very much enjoyable. **Xiaoming Huo** offered me key ideas and inspiration in tackling with my thesis problems, which I deeply appreciate. His critical reading and sharp comments made this dissertation more rigorous. My warm thanks is extended to the other committee members, **David Goldsman**, **Wenke Lee**, and **Joseph Wu** for their careful reading and valuable comments on this dissertation.

I also thank my friends, faculty, and staff at the School of Industrial and Systems Engineering program too many to name. Special thanks goes to **Terry Murphy** and **Jane Chisholm** for providing me with many useful tips during my studies.

Finally, I thank my family for their endless support throughout. Particulary, my special thanks goes to my wife, **Sun-Kyoung**, for her unselfish support through the whole years of study. Also, I am grateful for the love and concerns of my **parents** and **grandmother**. Completing the Ph.D. would have been impossible without their love and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

During the last decade, data mining has received great attention from various fields. This thesis investigates data mining problems in tree-based models and large-scale contingency tables. The first half of the thesis pertains to the tree-based models for the classification problem, which have been very popular in various fields because of their interpretability and flexibility. Tree modeling involves two major steps: tree growing and tree pruning. Tree growing searches over the whole data set to find the splitting point that leads to the greatest improvement in a specified score function. Once the trees are grown, tree pruning pursues the right sized tree that provides the best estimate of error when the tree is applied to unseen data. In this thesis, we propose a novel algorithm for tree pruning, called frontier-based tree pruning (FBP). The new method has an order of computational complexity comparable to cost-complexity pruning (CCP). Regarding tree pruning, FBP provides a full spectrum of information: namely, (1) given the value of the penalization parameter λ, it gives the decision tree specified by the complexity-penalization approach; (2) given the size of a decision tree, it provides the range of the penalization parameter λ, within which the complexity-penalization approach renders this tree size; and (3) it finds the tree sizes that are *inadmissible*, — so regardless of what the value of the penalty parameter is, the resulting tree, based on a complexity-penalization framework, will never have these sizes. Simulations on real data sets reveal surprising results: in the complexity-penalization approach, most of the tree sizes are inadmissible. FBP facilitates a more faithful implementation of cross validation (CV), which is favored by simulations. As an extension of the FBP algorithm, we study how CV performs in tree-based models. Considering the abundant results available on applying

CV to regression models, there is little research on the effects of CV in classification models due to their nonlinear structure. The main purpose of this study is to explore the behavior of CV in tree-based models. We report simulation studies that compare a cross-validated tree classifier with an oracle classifier that is ideally derived on the knowledge of underlying distributions. The main observation of this study indicates that the difference between the testing and training error from a cross-validated tree classifier and an oracle classifier empirically has a linear regression relation. The "slope" and the "R^2" of regression models are employed as the performance measures of a cross-validated tree classifier. Moreover, simulation reveals that the performance of a cross-validated tree classifier depends on the geometry, the parameters of the underlying distributions, and the sample sizes. Such observations can explain and justify the behavior of CV in tree-based models.

The second half of the thesis presents multiple testing in large-scale contingency tables and its application to pattern recognition of protein structures. One of the most common test procedures using two-way contingency tables is the test of independence between two categorizations. Current significant tests such as χ^2 or likelihood ratio tests provide overall independency but bring limited information about the nature of the association in the contingency tables. The main purpose of this study is to develop a follow-up method to χ^2 or likelihood ratio tests that can analyze the individual cells in the contingency table. We propose a framework of multiple testing procedures for testing independence of the cell categories in contingency tables. In the simulation study, we compare the power, type I error, and false discovery rate of five different testing procedures in the contingency table. We observe that no single procedure is superior for every scenario examined. In addition, we record the relationships among the proportion of true null hypotheses, power, type I error, and false discovery rate. Finally, we employ the proposed method to identify the patterns of pair-wise

associations between amino acids involved in β-sheet bridges of proteins. We identify a number of amino acid pairs that exhibit either strong or weak association. These patterns provide useful information for algorithms that predict secondary and tertiary structures of proteins.

CHAPTER I

INTRODUCTION

1.1. Motivation and Contribution

An unprecedented wealth of data has been generated from various fields. The huge demand for the analysis and interpretation of these data is being managed under the name of "data mining," or "knowledge discovery." The main purpose of data mining is to find useful information from a large and complex data set. However, since many of the current methods provide only limited solutions to complex situations, appropriate methods that overcome such limitations must be developed. Over the past several decades, numerous studies on data mining, including the development of new methods or extension of existing ones, have been performed in both academia and industry. This thesis discusses data mining methods and their applications.

1.1.1 Investigation of Tree-Based Models

In this thesis, we propose a novel algorithm for tree pruning and investigate the performance of cross validation in tree-based models. Tree-based models are very popular in the data mining community because they provide interpretable rules and logic statements that enable more intelligent decision making. Many popular data mining software packages include tree-based classification methods. One of them, CART, proposed by Breiman *et al.* (1984), has been implemented in S-Plus, Insightful Miner, CART in Salford systems, Enterprise miner in SAS, and many other statistical or data mining packages. Other tree-based methods such as MARS, TreeNet, and MART can be also found in Salford systems. In machine learning, C4.5 and C5.0

have been widely used. As their name suggests, tree-based methods generate tree-structured output from non-parametric input. From a geometrical point of view, tree-based methods partition the feature space into a set of rectangles and then fit a simple model to each one (Hastie *et al.*, 2001). The left panel of Figure 1 illustrates a simple tree structure. The tree has three layers of nodes. The circle in the first layer is the root node and the circle in the second layer is the intermediate node. The other three rectangles in the second and third layers are terminal nodes. The root and intermediate nodes have child nodes. However, the terminal nodes do not have child nodes by definition. The right panel of Figure 1 shows recursive partitioning. Two line segments (i.e., x1=c1 and x2=c2) separate the dots from the rectangle and thus generate three disjoint regions (A, B, and C). By this recursive partitioning, the nodes or regions become purer. Equation 1 is the representation of a tree-based

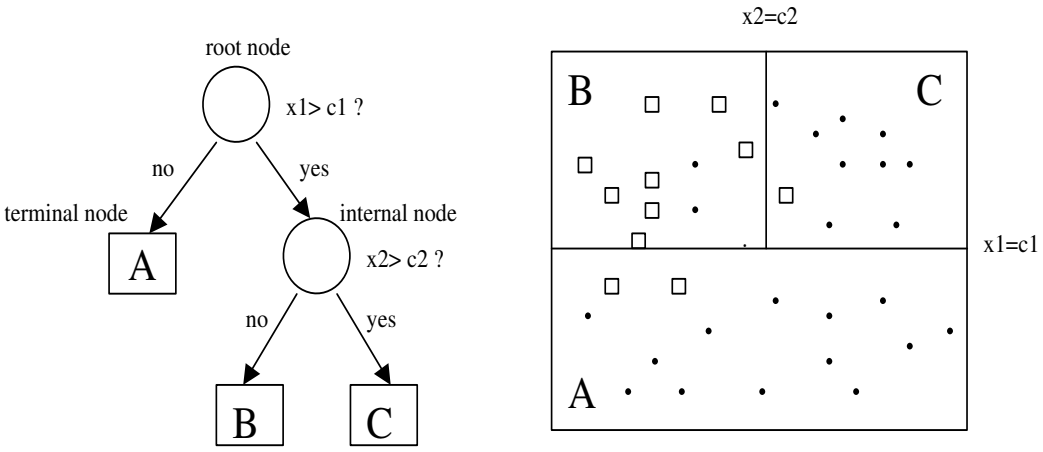

Figure 1: An illustration of tree structure. x1 and x2 are variables. Dots and rectangles show different classes.

model based on a regression model.

$$\hat{f}(X) = \sum_{i=1}^{3} c_i I\{(x_1, x_2) \in R_i\} \quad R_i \in \{A, B, C\}. \tag{1}$$

The most comprehensive study on tree-based methods was performed by Breiman

et al. (1984) in their book, *Classification and Regression Trees.* Minger (1989a, 1989b) described concerns with tree growing and tree pruning and compared a number of tree-based methods empirically. Numerous splitting rules such as information gain (Quinlan, 1987), Geni index (Breiman *et al.*, 1984), deviance (Clark and Pregibon, 1992), and others related to tree growing, have been proposed.

Breiman *et al.* (1984) pointed out that obtaining the right-sized tree is more important than selecting good split that most improves classification accuracy. In general, the full grown tree is not the best for classifying a new data set since it is overfit by the training set. There are two solutions to avoid this problem. The first solution is the direct stopping methods, which attempt to stop tree growing before it overfits the data. CHAID by Kass (1980), and other methods (Sethi and Sarvarayudu, 1982; Loh and Vanichsetakul, 1988) implemented this approach. Another solution is tree pruning, the process of removing leaves and branches to improve the performance of the decision tree (Berry and Linoff, 1997). The main goal of tree pruning is to find the right sized tree which minimizes the error rate when used to classify testing sets. Several research (Quinlan, 1993, Kim and Koehler, 1994) reveal that tree pruning is more effective than direct stopping methods. Various tree-pruning methods, listed in Table 1, have been proposed.

Table 1: List of tree-pruning algorithms

Methods	Developers	Strategy	Pruning set
Cost-complexity	Breiman *et al.* (1984)	Bottom-up	Yes
Reduced error	Quinlan (1987)	Bottom-up	Yes
Minimum error	Cestnik and Bratko (1991)	Bottom-up	Yes
Critical value	Mingers (1989b)	Bottom-up	Yes
Pessimistic	Quinlan (1987)	Top-down	No
Bootstrap-based	Crawford (1989)	Bottom-up	Yes
Error-based	Quinlan (1993)	Bottom-up	No
Minimum description length	Mehta *et al.* (1995)	Bottom-up	Yes
Dynamic programming based	Li *et al.* (2001)	Bottom-up	Yes
Frontier-based	**Huo, Kim, Tsui, Wang (2004)**	**Bottom-up**	**Yes**

In this thesis, we mainly focus on tree pruning and develop a new algorithm called frontier-based tree pruning. FBP utilizes the cost-complexity function (CPLF), which pursues the goal of tree pruning by finding the best compromise between error rate and tree size. FBP is similar to the cost-complexity pruning (CCP) in that both use the same CPLF but they are quiet different from an algorithmic point of view. Furthermore, FBP improves classification accuracy and provides a number of useful by-products.

Once we determine models (e.g., tree-based models), the next step is to judge the quality of the fitted model. Score functions are generally used to quantify how well a model structure fits a given data set. A simple generic score function is least squared error for a continuous response and the 0-1 classification rule for a categorical response. Once some form of score function is assigned, model selection or parameter selection can take place. Since models are described in terms of unknown parameter(s), identifying them is one of the most important tasks in data mining. Some score functions are amenable to mathematical manipulation (e.g., simple derivative), but a score function for classification problems is difficult to analytically minimize or maximize. Cross validation (CV) has been widely used in this situation and characterizing why and how a CV method works has been of paramount interest. In regression problems, the score function is continuous, so the behavior of CV can be easily studied. However, in classification problems, where the responses are categorical (i.e., discrete), equivalence between CV and some known existing criteria is difficult or even impossible to establish. The behavior of CV for categorical response is difficult to analyze due to their non-linearity. This thesis introduces an experimental approach that illustrates the behavior of the tree-based classifier selected by CV (referred to as the *"cross-validated tree classifier"*).

1.1.2 Multiple Testing in Large-Scale Contingency Tables: Application to the Pattern Recognition of the Protein Structure

The second part of this thesis presents multiple testing in large-scale contingency tables and its application to the recognition of protein structural patterns. One of the central issues of statistical analysis is to discover the significance of patterns through hypothesis testing. Traditional single-hypothesis testing finds the best critical region with the lowest type II error given an acceptable type I error (=α). Then all rejection regions are considered to have a type I error that is less than or equal to α. Type I and II errors are defined as follows:

$$\text{Type I error} = \min \ P[\tau \in \Gamma | H_0 \ \text{ true}],$$

$$\text{Type II error} = \min \ P[\tau \notin \Gamma | H_A \ \text{ true}],$$

where Γ is a rejection region and τ is the value of the test statistic.

When the experiment involves performing more than one test (a multiple testing problem), the situation becomes much more complicated. If we apply the same procedure to multiple testing problems as in a single testing problem, we may encounter many false positives because of the multiplicity problem. More precisely, this problem leads to an exponential increase of false positive rates as the number of hypotheses increases in multiple testing problems. In order to avoid this problem, many procedures that control the family of hypotheses are suggested. Excellent surveys were done by Schaffer (1995). Here, we present a brief and non-exhaustive summary of multiple testing procedures.

Let's express the ordered p-values for the n hypotheses being tested from the smallest to the largest: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(n)}$, and let $H_{(i)}$ be the hypothesis associated with the ith p-value, i.e., $p_{(i)}$.

The Simple Bonferroni Procedure

1. For a fixed α, where $0 \leq \alpha \leq 1$.

2. Reject H_i if $p_i \leq \alpha_i$, where $\alpha_i = \frac{\alpha}{n}$.

Holm's Procedure

Holm's procedure involving multi-stages is an extension of the Bonferroni procedure

.

1. For a fixed α, where $0 \leq \alpha \leq 1$.

2. $p_{(1)} \geq \frac{\alpha}{n}$?

   if yes, accept all hypotheses without further testing.

   if no, reject $H_{(1)}$ and consider the next hypothesis, $H_{(2)}$.

3. $p_{(2)} \geq \frac{\alpha}{n}$?

   if yes, accept all hypotheses $H_{(i)}$, for $i \geq 2$.

   if no, reject $H_{(2)}$ and consider the next hypothesis, $H_{(3)}$.

4. Continue the above steps until the first j such that $p_{(i)} \geq \frac{\alpha}{(n-j-i)}$.

   in tree-based classifiers.

Remark: When the test statistics are independent, both the Bonferroni and Holm's procedure can be improved by replacing $\frac{\alpha}{n}$ by $1-(1-\alpha)^{\frac{1}{n}}$, the Dunn-Sidak correction.

Hochberg's Procedure

Hochberg's procedure is a modified version of Holm's procedure which works backward, dealing with the largest p-value first.

1. For a fixed α, where $0 \leq \alpha \leq 1$.

2. $p_{(n)} \leq \alpha$?

   if yes, reject all hypotheses without further testing.

   if no, consider the hypothesis, $H_{(n-1)}$.

3. $p_{(i)} \leq \frac{\alpha}{2}$?

   if yes, reject all hypotheses $H_{(i)}$, for $i \leq n-1$.

   if no, compare $p_{(n-2)}$ with $\frac{\alpha}{3}$.

4. Generally, if $p_{(n-i)} \leq \frac{\alpha}{(n-i+1)}$, reject $H_{(i)}$ for $i \leq n-i$.

Hommel's Procedure

Hommel's procedure is more powerful than Hochberg's procedure although it is more complicated.

1. For a fixed α, where $0 \leq \alpha \leq 1$.

2. $\hat{i} = \max[i : p(n-i+j) > \alpha \cdot \frac{j}{i}]$ for $j = 1, 2, \cdots, i$.

3. Reject all hypotheses with $p_{(i)} \leq p_{(\hat{i})}$.

In general, the above multiple testing procedures are devised to work with two, three, or at most ten hypothesis tests at the same time. However, the advent of high technology has produced huge quantities of data, which has led to an increase in the number of hypotheses we need to consider simultaneously: 100, 1000, or even more than 10,000 tests. This has posed new and challenging problems for statisticians. To address this problem, Benjamini and Hochberg (1995) proposed the False Discovery Rate (FDR). The motivation of the FDR is that we may run a very large number of tests, and those declared significant would be subject to further study. For more details, see Chapter IV. FDR has been frequently used for microarray analysis to find co-expressed genes (Tusher *et al.* (2001), Efron *et al.* (2001), Efron and Tibshirani (2002), Dudoit *et al.* (2003)) as well as a genetic study to identify drugs causing mutations in the viral genome (Efron, 2004). Moreover, FDR has been applied to identify active voxels in neuroimaging data (Genovese *et al.* (2002) and Wink *et al.* (2004)). In these studies, a hypothesis test is performed in each voxel to determine

whether the voxel contributes to classification between different experimental conditions. As an extension of the original FDR , Storey (2002, 2003) and Storey *et al.* (2004) introduced the positive False Discovery Rate (pFDR) and Efron *et al.* (2001) proposed the Local False Discovery Rate (Local FDR). Moreover, the case when the hypotheses are dependent was considered by Yekutieli and Benjamini (1999) and Benjamini and Yekutieli (2001). In this thesis, we propose a multiple testing procedure for an statistical inference of independence in each cell of contingency tables. The multiple testing procedure in contingency tables overcome the limitations of the globally significant tests such as χ^2 and likelihood ratio tests in that it provides more information about the nature of the associations in each cell in the contingency tables. Moreover, the procedure has advantages over subjective methods such as normal probability plotting (Haberman, 1973) and partitioning of χ^2 (Lancaster, 1949). To illustrate the advantages, we use our proposed procedure to identify the pattern of β-strands formed by the associations of pair-wise amino acids. Knowledge of these patterns can improve the prediction accuracy of protein structures. For instance, the secondary prediction problem involves predicting the location of α-helices, β-sheets, and loops from a one-dimensional amino acid sequence. Current research reveals that many methods achieve relatively accurate prediction rate when identifying α-helices. However, prediction rate for finding β-sheets remains significantly low because of the pair-wise associations. Hence, identifying the pair-wise associations in β-sheets directly leads to improved prediction of secondary structure. In this thesis, we apply to the multiple testing procedure in contingency tables to the identification of the patterns of pair-wise association in β-sheets.

1.2. An Overview of Genomics and Proteomics

In this section we introduce some basic knowledge of biology to enhance understanding of Chapter IV.

1.2.1 Background of Molecular Biology

The study of genetics, as a set of principles and analytic procedures, did not begin until 1866, when Gregor Mendel performed a set of experiments that pointed to the existence of a biological element called the gene, the basic unit responsible for the transmission of a single characteristic. Until 1944, chromosomal proteins were generally assumed to carry genetic information with molecule deoxy-ribonucleic acid (DNA) playing a secondary role. This assumption was disproven by Avery and McCarty, who demonstrated that DNA was the major carrier of genetic material in a living organism. In 1953, James Watson and Francis Crick deduced the three-dimensional double helix structure of DNA and immediately posited its method of replication. In February 2001, a venture company, Celera, published the first draft of the human genome. Figure 2 describes the flow of genetic information in cells from DNA to RNA to protein. All cells, from prokaryote to eukaryotes, express their

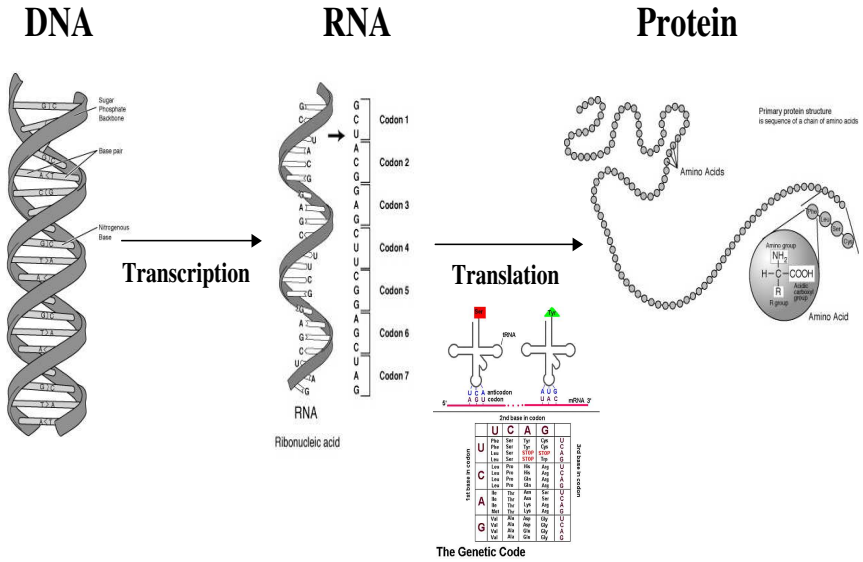

Figure 2: An illustration of central dogma.

genetic information in this way. This process has been referred to as "the central dogma of molecular biology." DNA is the double strands of nucleotides whose bases

are adenine (A), guanine (G), cytosine (C), and thymine (T). Complementary base pairs between two strands are held together by hydrogen-bonding within the helix structure. Since each strand of DNA contains a sequence of nucleotides that is exactly complementary to that of its partner strand, each strand can act as a template for the synthesis of a new complementary strand (Albert *et al.* 1997). The nucleotide A will pair with T and G with C. This process of synthesis describes the replication of DNA. As described earlier, the final destination of the central dogma is protein. However, DNA does not directly encode protein but rather acts as a controller for producing protein. When a particular protein is needed, appropriate parts of the DNA are transcribed onto another type of nucleic acid called ribonucleic acid (RNA). The chemical components of RNA are similar to those of DNA containing A, G, C, and U instead of A,G, C, and T. In spite of the substitution of U for T, the structures of DNA and RNA are significantly different. Whereas DNA has a double helix structure, RNA has only one strand. There are three different kinds of RNA: mRNA, rRNA, and tRNA. The main role of mRNA is to translate into protein. Three adjacent nucleotides in mRNA, called codon, specify one amino acid. tRNA can select the appropriate amino acid with the assistance of aminoacyl-tRNA synthetase and its anticodon pair with the codon in mRNA. Finally, the information of RNA is translated into a ribosome, consisting of ribosome protein and rRNA. tRNA with amino acid is combined with the codon in mRNA at the ribosome, and it produces a polypeptide chain otherwise known as a protein.

1.2.2 Proteins and Protein Structures

Most of the primary functions of cells are determined by proteins. These complex molecules exist in various forms that allow them to perform a variety of activities that are essential to life. A fundamental principle of protein science is that protein structure leads to protein function. Proteins that perform similar functions tend

to show a significant degree of structural homology. Therefore, understanding the protein structure is a key step to revealing the protein function. Because protein functions are diverse and inferred from the protein structure, we can easily deduce that the protein structure is also diverse. Thus, predicting the protein structure is a challenging task. Ample research has been devoted to identifying regularities and patterns of protein structure. In general, the structure of a protein is described by four levels of classification that facilitate description and understanding of proteins. The four structural levels are primary, secondary, tertiary, and quaternary.

1.2.2.1 The Primary Structure of Proteins

Protein is a polymeric compound made of 20 amino acids listed in Table 2. Figure 3 is a simple illustration of two amino acids with a peptide bond.

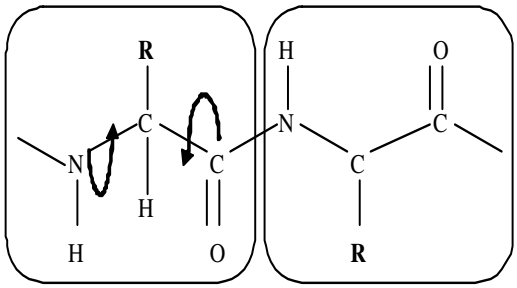

Figure 3: The structure of two amino acids in a polypeptide chain.

The R group is different for each of the 20 amino acids. Neighboring amino acids are joined by a peptide bond between the C=O and NH groups. Therefore, the N-C_α-C sequence is repeated throughout the protein, forming the backbone of the three-dimensional structure. The conformation of the protein backbone in space is determined by the angles of these bonds represented the twisted arrows in Figure 3.

Table 2: The building blocks of proteins: 20 amino acids

Amino acids	Three letter (One letter)	Amino acids	Three letter (One letter)
alanine	Ala(A)	methionine	Met(M)
cysteine	Cys(C)	asparagine	Asn(N)
aspartic acid	Asp(D)	proline	Pro(P)
glutamic acid	Glu(E)	glutamine	Gln(Q)
phenylalanine	Phe(F)	arginine	Arg(R)
glycine	Gly(G)	serine	Ser(S)
histidine	His(H)	threonine	Thr(T)
isoleucine	Ile(I)	valine	Val(V)
lysine	Lys(K)	tryptophan	Thp(W)
leucine	Leu(L)	tyrosine	Tyr(Y)

1.2.2.2 The Secondary Structure of Proteins

The secondary structure is the representation of amino acids as specific forms such as α-helices, β-sheets, and loops (Figure 4). The physicochemical properties of amino acids and other environmental factors determine the way in which they are arranged in space relative to each other.

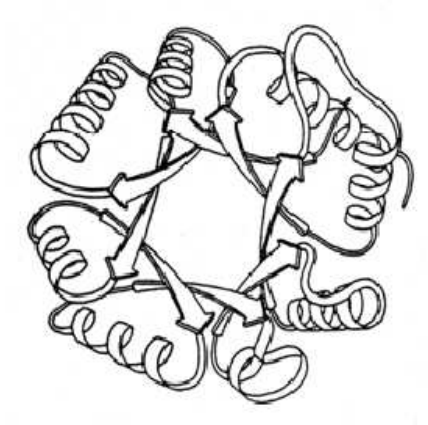

Figure 4: Secondary structure of proteins: α-helices (corkscrew staircase), β-sheets (big arrow), loops (line).

α-helices

α-helices are usually formed with a hydrogen bond between the ith and the $(i+n)$th amino acid residues, where n is in general 3, 4, or 5. Thus, its shape is similar to a corkscrew staircase. This structure is very stable but flexible; therefore, it is often seen in parts of a protein that need to bend or move. Figure 5 illustrates an α-helix.

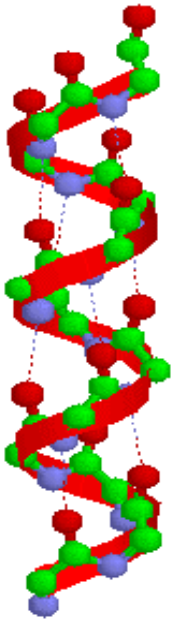

Figure 5: α-helix structure: The α-helix is stabilized by internal hydrogen bonds shown here as dashed lines.

β-sheets

Unlike α-helices, which are built up from one contiguous region of the polypeptide chain, β-sheets (Figure 6) are more complex, resulting from a combination of several disjoint regions, called β-strands. β-strands are typically five to ten residues long. In the folded protein, these strands are aligned adjacent to each other in parallel or antiparallel fashion. In parallel sheets, the strands are arranged in the same direction with respect to their amino terminal (N) and carboxy terminal (C) ends. In the antiparallel sheets, the strands alternate their amino and carboxy terminal ends such

that a given strand interacts with the other strands in the opposite orientation. A β-strand can have one or two partner strands, and individual amino acid can have zero, one, or two hydrogen bonds with one or two residues in a partner strand. Hydrogen (H) bonds between parallel and antiparallel strands have distinctive patterns, but the exact nature and behavior of the β-sheet long-range interaction is not clear (Baldi *et al.*, 2000).

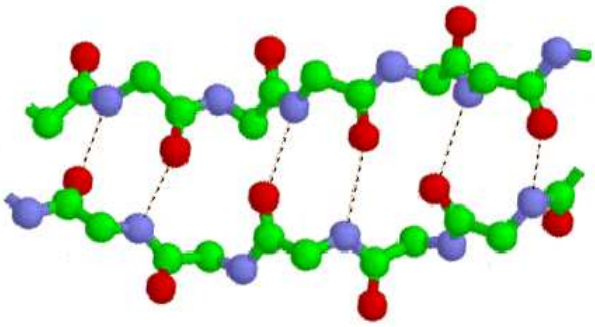

Figure 6: β-sheets structure: The β-sheet is stabilized by hydrogen bonds, shown as dashed lines.

Loops or Coils

The majority of secondary structures consist of α-helices and β-sheets. These regular structures are connected with some irregular structures such as loops or coils. They are usually comprised of small residues (e.g., proline, glycine) and often responsible for sharp bends and twists in α-helices and hair-pins in β-sheets. Because of their irregularity, they are hard to predict.

1.2.2.3 The Tertiary Structure of Proteins

The tertiary structure, shown in Figure 7, refers to the three-dimensional structure of the entire polypeptide chain. The tertiary structure is stabilized by hydrogen bonding between individual amino acid residue and hydrophobic forces. As helices and sheets

are units of secondary structure, so is the domain a unit of tertiary structure. Domains can be considered to be a segment of a polypeptide that folds independently of other segments (Brown, 2002). Each domain can be described by its fold. While some proteins consist of a single domain, others consist of several or many.

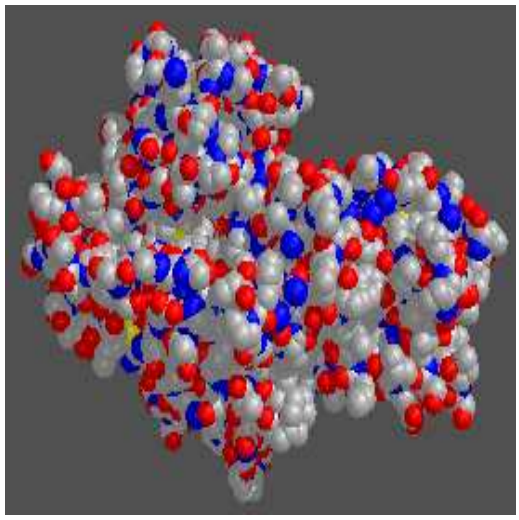

Figure 7: The tertiary structure of proteins.

1.2.2.4 The Quaternary Structure of Proteins

The quaternary structure, shown in Figure 8, is the structure resulting from the association of two or more polypeptides, each folded into its tertiary structure. A quaternary structure explains complex functions, including several involved in the genome expression, even though not all proteins form a quaternary structure. Some quaternary structures are held together by disulfide bridges between different polypeptides. However, many proteins comprise looser associations of subunits stabilized by hydrogen bonding and a hydrophobic effect similar to those of β-strands (Brown, 2002).

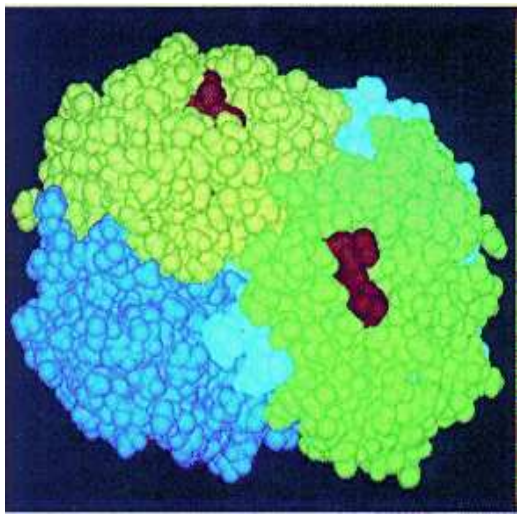

Figure 8: The quaternary structure of proteins.

1.3. Outline of the Thesis

This thesis is composed of two parts. The first part deals with an investigation of tree-based models. The second part studies the multiple testing in large-scale contingency tables and its application to the pattern recognition of protein structures.

The organization of Chapter II is as follows: Section 2.2 presents the key motivation of the algorithmic design. Section 2.3 reviews the complexity-penalized tree-pruning algorithm. Section 2.4 describes the proposed FBP algorithm. Section 2.5 describes how FBP can be integrated with CV, and how the CV in FBP is different from the CV in CCP. A study of the stability of CV will be given in this section too. Section 2.6 presents simulations. Section 2.7 describes the architecture of the FBP algorithm. Some concluding remarks are made in Section 2.9.

Chapter III is organized as follows: Section 3.2 reviews the cross-validation principle. Section 3.3 describes the FBP method and its links to cross-validation. Section 3.4 presents some initial theoretical analysis. Section 3.5 describes the simulation results. The final section, Section 3.6, completes the chapter with a few concluding remarks and suggestions for future study.

Chapter IV is composed of the following sections: Section 4.2 reviews some of the multiple testing procedures. Section 4.3 presents the multiple testing procedure for the contingency tables, the main topic of this Chapter. Section 4.4 presents the simulation studies, which compare the power, type I error, and false discovery rate of several multiple testing procedures in contingency tables. Section 4.5 presents the applications, which describe the identification of significant amino acid pairs in β-sheet bridges. Finally, Section 4.6, contains concluding remarks.

Chapter V summarizes the thesis and presents conclusions.

CHAPTER II

A FRONTIER-BASED TREE-PRUNING ALGORITHM (FBP)

2.1. Introduction

Tree-based methods, due to their flexibility and interpretability, have gained enormous popularity in statistical modeling and data mining. Li *et al.* (2001) give an excellent survey on tree building (including tree growing and tree pruning). An important technical question in building a statistically optimal tree is to find an optimal strategy to *prune* a huge tree.

This Chapter is focused on the methodology of tree pruning. The concept of *complexity-penalization* is well adopted in this topic. The essence of this approach is to solve

$$\min_{T} \quad L(T) + \lambda |T| \tag{2}$$

where

- T denotes an *admissible* subtree, which can correspond to a partition of the state space,
- $L(T)$ is the error rate of the corresponding decision tree,
- λ is the penalization parameter, which was mentioned earlier, and
- $|T|$ denotes the size of the tree, normally the number of *leaves* (i.e., terminal nodes) in tree T.

Such an approach was essentially the objective in cost-complexity pruning (CCP) (Breiman *et al.*, 1984). In the tree-pruning framework, there are two interwoven quantities, the value of the penalization parameter λ, and the size of the tree. While considering the performance of a tree model on testing data, one needs to consider *generalization error*. We intentionally leave it out in the above bulleted list, because it is not essential in designing the algorithm. It eventually will be used in evaluating the obtained decision tree.

In pruning a large tree, the following four questions are of interests:

1. Given the value of the penalization parameter, applying the criterion of "minimizing the complexity-penalized loss function" (which will be elaborated later), what will be the size of the pruned tree?

2. Given the target tree size, what is the range of the penalization parameter, λ, so that when the principle of "minimum complexity-penalized loss function" is applied and the parameter λ is in this range, the size of the pruned tree is equal to this target size?

3. Are there some sizes of trees for which no value of the penalization parameter will render pruned trees that are of these sizes? We will call these occasions *inadmissible*.

4. Given the unseen data, does the pruning method help improve classification accuracy? In other words, can the generalized error be reduced? This will be addressed by combining with cross validation (CV).

The first three questions are related to an algorithmic parameter. The last one is on the generalization error. Answering these questions evidently gives users insight on which tree model should be chosen.

The key contribution of this Chapter is a different algorithm for tree pruning. Apart from the local greed approach adopted in CCP (Breiman *et al.*, 1984), the proposed method *Frontier-based pruning* (FBP) keeps *all* the useful information and propagates it in a bottom-up fashion to the top layer (i.e., the root node). A specific algorithm is designed so the above can be achieved efficiently — having nearly the same order of complexity as CCP.

The proposed algorithm can automatically and simultaneously answer the first three of the above four questions. Identification of inadmissible tree sizes has not been considered in any other tree- pruning methods, although it seems to be observed — see the justification of a dynamic-programming approach in Li *et al.* (2001). To our knowledge, this is the first time it is explicitly addressed in a *quantitative* manner. The proposed method has a very nice graphical interpretation which has a connection to pareto-optimality, and is highly intuitive.

To prune a single tree, both CCP and FBP solve the same problem, which is to minimize the objective in (2). However, because FBP provides an entire spectrum of information on pruning, it facilitates a more faithful realization of the cross-validation principle. In simulations, it is shown that such a difference can lead to improvement in testing errors.

Also due to the design of FBP, one can study whether a CV is stable. More specifically, one may ask whether the CV partitioning can significantly affect the output of CV. We study this problem through simulation. Because of the availability of FBP, a nice illustration can be generated.

2.2. The Main Idea of the Algorithm

An illustration of the main idea of the proposed approach is Figure 9. The value of λ is considered the cost of an additional node in a tree. For a given λ, when the target size of the tree is m, the minimum value of the complexity-penalized loss function

(CPLF) is $c_m + m\lambda$, where c_m is a constant (the intercept). The y-axis is the value of the CPLF. Given λ, the slope m of $c_m + m\lambda$ is the size of the subtree. Define $f(\lambda) = \min_{m=1,2,\ldots}\{c_m + m\lambda\}$. For a given value λ_0, the slope of $f(\lambda_0)$ is the size of the optimal subtree. It is not hard to see that $f(\cdot)$ is piecewise linear. Given a fixed integer (denoted by m_0), there is an interval of λ within which the slope of $f(\cdot)$ is equal to m_0. In other words, if λ takes a value inside this interval, the size of the optimal subtree is m_0. To check if a specific tree size m_0 is optimal, one can check if $c_m + m_0\lambda$ composes the part of the curve that is associated with $f(\cdot)$.

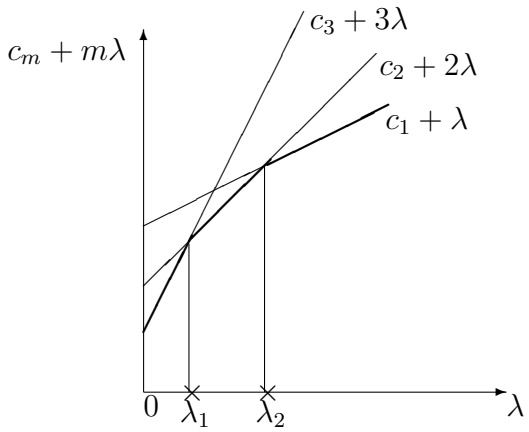

Figure 9: An illustration of the frontier-based tree-pruning algorithm.

A lower bound in a bundle of lines in a graph is associated with the size of an optimal subtree (within the domain of the parameter λ); this is analogous to an efficient frontier in investment theory (Luenberger, 1998), a curve representing a set of efficient portfolios that maximize expected returns at each level of portfolio risks; see Markowitz (1952). This interpretation illustrates why our method is called frontier-based tree pruning.

In tree pruning, the key problem is that given two sets of lines, how do we find their common lower bound? More challengingly, how do we achieve this with the lowest possible computational cost? Our algorithms (Section 2.4) provide answers.

2.3. Tree Pruning

Tree methods in data analysis and modeling were "boosted" by Breiman *et al.* (1984). Since then, a tremendous literature has been developed. Li *et al.* (2001) indicate that in building a tree, the most important part is to prune a redundant tree. In this Chapter, we adopt this conclusion. Tree methods have been extremely successful in data mining. For example, the most popular data mining tools — CART and MARS (Salford Systems, 2004) — are based on tree methods. Various issues in building an "optimal" tree have been studied; see Buja and Lee (2001), for example. Here, we assume that a big and redundant tree has been built.

The complexity-penalized tree-pruning approach was described in Breiman *et al.* (1984). The idea is originally rooted in statistical model selection. A significant advantage of adopting this principle is that people can prove various optimality results, e.g., Donoho (1997, 1999), etc.

In the rest of this section, we first review the general principle of complexity-penalized tree pruning (Section 2.3.1). A bottom-up tree-pruning algorithm is described in Section 2.3.2.

2.3.1 The Principle of Minimizing a Complexity-Penalized Loss Function

We give more detail here on a previously mentioned idea. The objective of a complexity-penalized tree-pruning approach (from a big redundant tree) is to find a subtree that minimizes the CPLF. The principle of minimizing the CPLF will be described in the following. Let T denote a big un-pruned tree. In tree modeling, each tree is associated with a *recursive dyadic partitioning* (RDP) of a state space. Without loss of generality, we focus our attention on binary trees. Due to their simplicity and interpretability, binary trees are often used in tree-based algorithms. The size of a tree (or a subtree) is the number of *terminal* nodes, which is also equal to the number of regions in an RDP of the state space. If two regions in an RDP are merged, the

size of the tree is reduced by one. In tree pruning, the region merging is equivalent to pruning a two-leaf branch from a binary tree. A subtree is a tree that can be obtained by repeating the above procedure. A subtree is denoted by T_b, where b is an index. Since each subtree is associated with a partition of the state space, one can choose the tree that minimizes a criterion function. In this case, this criterion function is the CPLF, which is defined in the following. Let $L(T_b)$ denote the loss function associated with tree T_b. Let $|T_b|$ denote the size of the tree T_b. The CPLF is $L(T_b) + \lambda|T_b|$, where λ is a penalizing parameter. The principle of minimum CPLF is to choose a subtree T_b that minimizes the CPLF. In other words, it is to solve

$$T_{b_0} = \underset{T_b}{\operatorname{argmin}}\{L(T_b) + \lambda|T_b|\}. \tag{3}$$

2.3.2 Bottom-Up Tree-Pruning Algorithm

A well-known algorithm can efficiently solve the above problem, a *bottom-up tree-pruning algorithm*. This algorithm uses the same idea as in other algorithms such as *best basis* (Coifman and Wickerhauser, 1992), which originated in the wavelet literature. This algorithm is also described in Breiman *et al.* (1984). Figure 10 illustrates the idea.

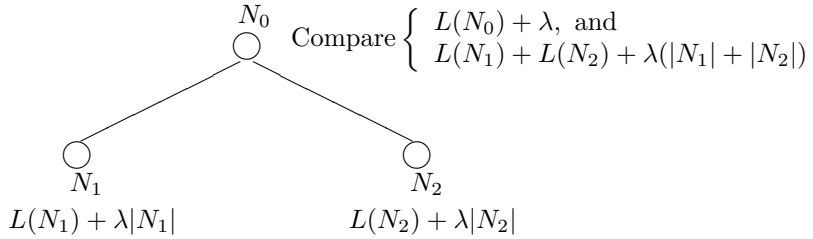

Figure 10: An illustration of bottom-up tree pruning.

We consider two terminal nodes N_1 and N_2 in a binary tree. Their common parent is N_0. Let $L(N_0)$, $L(N_1)$, and $L(N_2)$ denote the values of the loss function for the nodes N_0, N_1, and N_2, respectively. The CPLF associated with N_1 and N_2 are $L(N_1) + \lambda$ and $L(N_2) + \lambda$, respectively. Recall that in a binary tree, each node is associated with

a region in an RDP of the state space. When one goes up to the node N_0, there are two possibilities: splitting the region into two regions N_1 and N_2, or no splitting. If the region N_0 is split, then the value of the CPLF should be $L(N_1) + L(N_2) + 2\lambda$; otherwise, it should be $L(N_0) + \lambda$. Which one should be chosen depends on which value of the CPLFs is smaller. In a bottom-up tree-pruning algorithm, one starts at the bottom of a tree (or terminal nodes), and then repeatedly applies the above procedure until the top (root) node is reached. Note that when the nodes N_1 and N_2 are not terminal nodes, the expressions of their penalization functions are different. They should be $\lambda|N_1|$ and $\lambda|N_2|$. In the parent node (N_0), the comparison is between

$$L(N_1) + L(N_2) + \lambda(|N_1| + |N_2|), \qquad \text{(splitting)}$$

and

$$L(N_0) + \lambda, \qquad \text{(no splitting)}$$

which are also indicated in Figure 10. The bottom-up tree-pruning algorithm finds the subtree that minimizes the CPLF (Breiman *et al.*, 1984).

One advantage of the bottom-up tree-pruning algorithm is that it is computationally efficient. If the un-pruned tree has size $|T|$, it takes no more than $O(|T|)$ operations to find the minimizer of the (9).

The bottom-up tree-pruning approach has some intrinsic links with other approaches. For example, in Breiman *et al.* (1984) and Li *et al.* (2001), a method based on analyzing the reduction of the total loss function is discussed. This method is called CCP. Apparently, in the bottom-up tree-pruning algorithm, the choice of every step depends on whether the reduction of the loss function (following splitting the parent node, which is equal to $L(N_0) - L(N_1) - L(N_2)$) is larger than the value of the parameter λ (or the reduction of the penalization function $\lambda(|N_1| + |N_2| - 1)$ when the nodes N_1 and N_2 are nonterminal). This leads to the discussion of a pruning algorithm in the original CART book (Breiman *et al.*, 1984).

2.4. Frontier-Based Tree-Pruning Algorithm

There are five subsections. Section 2.4.1 describes the frontier-based tree-pruning algorithm. Section 2.4.2 interprets the occurrences of inadmissible tree sizes. Section 2.4.3 provides a fast algorithm in numerically realizing the idea in Section 2.4.1. Section 2.4.4 gives the computational complexity of our approach. Finally, Section 2.4.5 explains the connection between the frontier-based method and the dynamic-programming-based method.

2.4.1 Algorithm

Now we start describing the frontier-based pruning approach. The key point is to create a list of linear functions that have the form $c + m\lambda$ at each node, in which c is the value of a loss function and m is the size of a subtree. At the root node, the information is summarized.

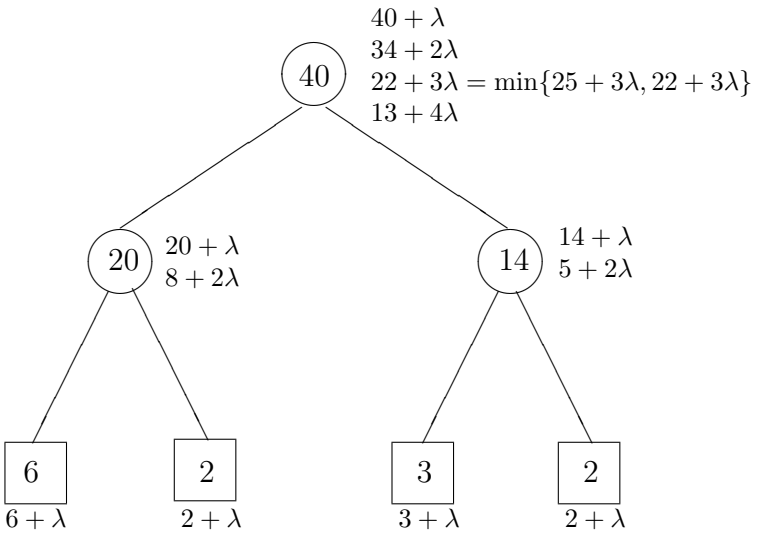

Figure 11: An example of the frontier-based tree-pruning approach.

To explain the FBP algorithm, we use an example in Figure 11 in which a circle indicates an intermediate node and a square indicates a terminal node. The values in the circles and squares are the values of the loss function. Let λ denote the penalizing parameter. At each node, all possible expressions of the CPLF are listed as the linear

functions of λ. For example, at all terminal nodes, the CPLFs have the form $c + \lambda$, where c is the value of the loss function. When one goes up in the tree, at an intermediate node, the expressions for the CPLFs have forms $c_m + m\lambda$, $m = 1, 2, \ldots$, where c_m is the value of the loss function when the size of the tree is m. Each node will have a list of linear functions. At each node, we need only determine the sequence of c_m's. The list at the parent node can be derived from the list at the two siblings. For example, for the node where a value 20 is inside a circle, the value of c_1 should be 20, and the value of c_2 should be $6 + 2 = 8$. There should be two linear functions at this node. Sometimes the value of the intercept c_m is not uniquely defined. For example, at the root node of the above example, when the size of the tree is 3, the intercept should take the minimum value between $25 = 20 + 5$ and $22 = 8 + 14$. We start from the bottom of the tree, and the lists of linear functions are built (following a bottom-up strategy) for all nodes.

The list building, described above, is the first step of the FBP algorithm. Consequently, the list at the root node is processed in the following way: in a Cartesian plane, the x-axis is taken to be the value of λ, and the y-axis is the value of the linear functions $c_m + m\lambda$. All the linear functions are plotted on this plane. An illustration of these linear functions is in Figure 9. The lower bound of these functions is considered. This function is

$$f(\lambda) = \min_{m=1,2,\ldots,T} \{c_m + m\lambda\},$$

where T is the number of terminal nodes, and $c_m + m\lambda$ are linear functions at the root node. The following observations indicate how to use f:

1. For fixed λ, the value $f(\lambda)$ gives the minimum CPLF.

2. For a fixed size of the tree m_0, let (a_0, b_0) denote the interval of the parameter λ such that when λ takes a value inside this interval, the slope of the function

$f(\lambda)$ is m_0. On the other hand, to get a subtree that has m_0 terminal nodes, the value of λ should be chosen in the interval (a_0, b_0).

3. It is possible that for an integer m_0, there is no part of curve $f(\lambda)$ such that its slope is m_0. In this case, the tree size m_0 is inadmissible — no matter what the value of λ is, in the minimum CPLF approach, the final tree size will never be m_0. An illustration of this case is in Figure 12.

2.4.2 Inadmissibility

It is possible that for a certain tree size, regardless of λ, the minimum CPLF approach will not render a tree that is of this size. In such a case, this tree size is called *inadmissible,* which is analogous to inadmissible estimators in statistics. Apparently the definition of *inadmissibility* relies on the minimum CPLF procedure. Even if a tree size is inadmissible, this does not necessarily imply that the tree of this size is not optimal with respect to other criteria. We would like to emphasize the dependence of this definition on the CPLF, and warn about the possible confusion between *admissibility* and *optimality.*

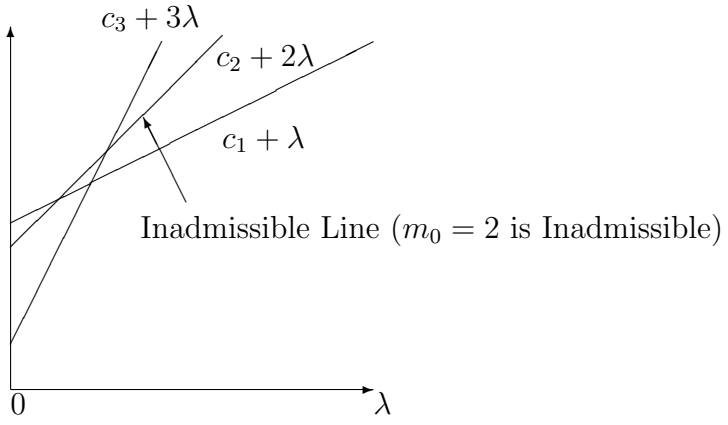

Figure 12: An inadmissible case.

An illustration of an inadmissible case is in Figure 12. In the FBP algorithm, inadmissible cases can be quickly identified. A fast numerical algorithm, which identifies admissible tree sizes as well as associated intervals, is presented in Section 2.4.3.

For the example in Figure 11, the tree sizes 2 and 3 are inadmissible, because

$$\begin{aligned} 34 + 2\lambda &\geq \min(40 + \lambda, 13 + 4\lambda), \qquad \text{and} \\ 22 + 3\lambda &\geq \min(40 + \lambda, 13 + 4\lambda). \end{aligned}$$

So the linear functions $34 + 2\lambda$ and $22 + 3\lambda$ are dominated by linear functions $40 + \lambda$ and $13 + 4\lambda$.

2.4.3 Algorithm to Find a Lower Bound in a Bundle of $a\lambda + b$ Lines

An algorithm that finds the lower bound of a bundle of lines having the form $k\lambda + c_k$ is described here. Here, slope k is a positive integer, λ is a variable, and the c_k's are the intercepts. When the number of lines is N, this algorithm finds the lower bound with at most $O(N)$ operations.

Formation. Suppose the lines are $\lambda + c_1$, $2\lambda + c_2$, $3\lambda + c_3$, $\ldots$, $N\lambda + c_N$. The lower bound of them is the function $f(x) = \min_{1 \leq k \leq N}\{k\lambda + c_k\}$. Apparently, $f(x)$ is a piecewise linear function, which is determined by the positions where the slope changes, and the constant slope within each interval. Recall that in tree pruning, we have $c_1 > c_2 > c_3 > \cdots > c_N$.

Algorithm. We start with the following table.

slope:	1
positions:	0

This indicates that there is one interval: $(0, +\infty)$. Within this interval, the slope is 1.

Now we take into account the line $2\lambda + c_2$. A new table is established,

slope:	2	1
positions:	0	$x_{1,2}$

where $x_{1,2}$ is the intersecting position of lines $\lambda + c_1$ and $2\lambda + c_2$. This is done by inserting a slope 2 at the beginning of the *slope* row and an intersecting position $x_{1,2}$ at the end of the *positions* row.

To illustrate the idea, now consider adding the line $3\lambda + c_3$. Recall that if

$$3x_{1,2} + c_3 < 2x_{1,2} + c_2, \tag{4}$$

then the line $3\lambda + c_3$ is lower than the line $2\lambda + c_2$ at the position $x_{1,2}$. Hence the line $2\lambda + c_2$ is always above the minimum of the line $\lambda + c_1$ and the line $3\lambda + c_3$. In other words, line $2\lambda + c_2$ is dominated; we do not need to consider the line $2\lambda + c_2$. Based on the above, a new table should be

slope:	3	1
positions:	0	$x_{1,3}$

where $x_{1,3}$ is the position where $\lambda + c_1$ and $3\lambda + c_3$ intersect. If the inequality (4) is false, the line $2\lambda + c_2$ is not dominated. The new table should be

slope:	3	2	1
positions:	0	$x_{2,3}$	$x_{1,2}$

which indicates that in intervals $(0, x_{2,3})$, $(x_{2,3}, x_{1,2})$, and $(x_{1,2}, +\infty)$, the slopes of the lower bound are $3, 2$, and 1, respectively.

In general, suppose that after step $n-1$ the table is

slope:	i_1	i_2	$\cdots$	i_k	1
positions:	0	x_{i_1,i_2}	$\cdots$	x_{i_{k-1},i_k}	$x_{i_k,1}$

where the k is the number of remaining lines. Consider adding the line $n\lambda + c_n$. It is not hard to verify that the set of points

$$\begin{aligned}
&(x_{i_1,i_2}, \quad i_1 x_{i_1,i_2} + c_{i_1}),\\
&(x_{i_2,i_3}, \quad i_2 x_{i_2,i_3} + c_{i_2}),\\
&\qquad\qquad\vdots\\
&(x_{i_{k-1},i_k}, \quad i_{k-1} x_{i_{k-1},i_k} + c_{i_{k-1}}),\\
&(x_{i_k,1}, \quad i_k x_{i_k,1} + c_{i_k}),
\end{aligned}$$

are concave. Hence there exists an integer l, such that when $j \le l$, we have

$$n\lambda + c_n \le i_j x_{i_j,i_{j+1}} + c_{i_j}; \tag{5}$$

and when $j > l$, we have

$$n\lambda + c_n > i_j x_{i_j,i_{j+1}} + c_{i_j}. \tag{6}$$

Hence the lines $i_j\lambda + c_{i_j}, j = 1, 2, \ldots, l$ are dominated. Hence the new table should be

slope:	n	i_{l+1}	i_{l+2}	$\cdots$	i_k	1
positions:	0	$x_{n,i_{l+1}}$	$x_{i_{l+1},i_{l+2}}$	$\cdots$	x_{i_{k-1},i_k}	$x_{i_k,1}$

The above procedure is repeated until all lines are added. The final table determines the configuration of the lower bound.

Cost of the algorithm. Obviously, it takes constant numbers of operations both to compute an interesting position and to carry out comparisons like (4), (5), and (6). Each line will be added once, and will be eliminated at most once. Thus, the overall cost of this algorithm is at most $O(N)$.

2.4.4 Computational Complexity

In a binary tree, suppose that the number of nodes and terminal nodes are equal to N and T, respectively. Then the number of operations required for FBP is no more

than $N + T(T-1)$. Moreover, in a binary tree, $N = 2T - 1$. Therefore, the order of complexity is $N + \frac{1}{4}(N^2 - 1)$ for the minimum CPLF algorithm with a fixed λ. But this is the price to pay for more information.

Theorem 2.4.1 *Let N and T denote the number of nodes and terminal nodes in a binary tree. If we use N to express the complexity, it takes no more than $N + \frac{1}{4}(N^2 - 1)$ operations to generate the lists of linear functions at all nodes.*

Proof. There are two stages in our proof. First, at all nodes, for terms like $c + \lambda$, it takes N operations to create them. Second, for terms like $c + m\lambda$, where $m \geq 2$, it can be shown that it takes no more than $T(T-1)/2$ operations to calculate all the related linear functions, and it takes no more than $T(T-1)$ operations to generate the lists. The second statement can be proved by induction. Here we explain the idea.

Assume that the second statement above is true for any tree with number of terminal nodes smaller than T. For a binary tree with T terminal nodes, let m_1 (resp. m_2) denote the number of terminal nodes that have the left (resp. right) child of the root node as an ancestor. Note here we follow the common terminology of a classification and regression tree. We must have $T = m_1 + m_2$. Based on the assumption, for each binary tree whose root node is one of the immediate children of the root node in the original tree, it takes $m_1(m_1 - 1)/2$ and $m_2(m_2 - 1)/2$ operations to compute all the linear functions. For the root node, it takes $m_1 m_2$ operations to compute all the necessary linear functions. Altogether, the number of operations to compute linear functions is

$$m_1(m_1 - 1)/2 + m_2(m_2 - 1)/2 + m_1 m_2 = (m_1 + m_2)(m_1 + m_2 - 1)/2 = T(T-1)/2.$$

Note that to find the minimum values, in some cases, it takes no more than the steps needed to scan through all the sums. Hence it requires no more than $T(T-1)/2$

operations. From all the above, the number of operations to create the lists of linear functions *at all nodes* is no more than $T(T-1)$, which is equivalent to $\frac{1}{4}(N^2-1)$.

The summary of the above two facts leads to the proof of the theorem. □

2.4.5 Connection with the Dynamic-Programming-Based Approach

In Li *et al.* (2001), a dynamic-programming-based pruning (DPP) is proposed. The advantage is that the DPP can find optimal trees with any sizes. In our FBP algorithm, by tracing back the route of the generating linear function $c_m + m\lambda$, one can extract the optimal size-m subtree. It can be shown that if the optimal subtree is unique and admissible, then both approaches will find the optimal one.

Based on the previous description of FBP algorithm, tracing back is straightforward. Figure 11 shows an example. Suppose one wants to find the optimal subtree that has three terminal nodes. The corresponding linear function at the root node is $22 + 3\lambda$, which is induced by $(8 + 2\lambda) + (14 + \lambda)$. Moreover, the linear function $8 + 2\lambda$ is made by $(6 + \lambda) + (2 + \lambda)$. Every time a linear function having the form $c + \lambda$ is reached, a terminal node is reached. Hence in this case, the optimal size-three subtree is made by the first two terminal nodes (from the left) and the intermediate node with misclassification rate being 14. The above procedure determines a subtree with size three. As mentioned in Li *et al.* (2001), it is possible that the optimal subtree is not unique. In that case, one can design some ad-hoc rules to specify it.

We now describe the consistency between the two approaches.

Theorem 2.4.2 *Among the subtrees that have size m, the one with the smallest misclassification rate can be found by both DPP and FBP.*

Proof. We prove only the case when the optimal subtree is unique and the goodness of fit is measured by the number of misclassifications. In DPP, it is known that the algorithm finds the subtree with the smallest number of misclassifications.

In FBP, the algorithm also finds the subtree with the fewest misclassifications. This can be proved by showing that if there is a subtree with fewer misclassifications, then at the root node, this subtree will generate a linear function with smaller intercept. This is contradictory to the definition in the FBP algorithm. From all the above, the DPP and FBP should generate the same subtree. □

2.5. Integration with Cross Validation

An nice feature of the FBP algorithm is that it can be integrated with CV. We begin with the principle of CV, then describe how the FBP can be used in implementing the CV.

Suppose the observations are $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_M, y_M)\}$, where M is the number of observations, the x_i's are predictor variables, and the y_i's are responses. Note that the x_i's can be multivariate. Suppose the above set is (equally) partitioned into K non-intersecting subsets: $S_1 \cup S_2 \cup \cdots \cup S_K$. At each step, we leave out one subset (say S_i) and use the remaining subsets to grow a tree and then prune a tree; the lower-bound function will be denoted $f_{-i}(\cdot)$. For each value of the parameter λ, the size of the optimal subtree and the subtree itself can be extracted. The optimal subtree determines a model. This model is then applied to the omitted subset (which is S_i). This is called *testing*. The error rate in testing can be computed and is denoted by $e_{-i}(\cdot)$. Note that functions $f_{-i}(\cdot)$ and $e_{-i}(\cdot)$ are of the same variable. Since $f_{-i}(\cdot)$ is piecewise linear, it is not hard to prove that $e_{-i}(\cdot)$ is piecewise constant (i.e., a step function). The principle of CV is to find the value of the parameter λ such that the average of the e_{-i}'s, $\sum_{i=1}^{K} e_{-i}/K$, is minimized. Throughout this Chapter, the above quantity will be called *cross-validation error* (CVE).

This principle can be easily implemented with FBP. The generation of functions $f_{-i}(\cdot)$ is already in FBP. The generation of functions $e_{-i}(\cdot)$ can be easily implemented. A simulation example for the Cleveland Heart Disease data (Blake and Merz, 1998)

is presented in Figure 13. Based on this, we conclude that the model made by using λ between 1 and 1.7 gives the minimum CVE.

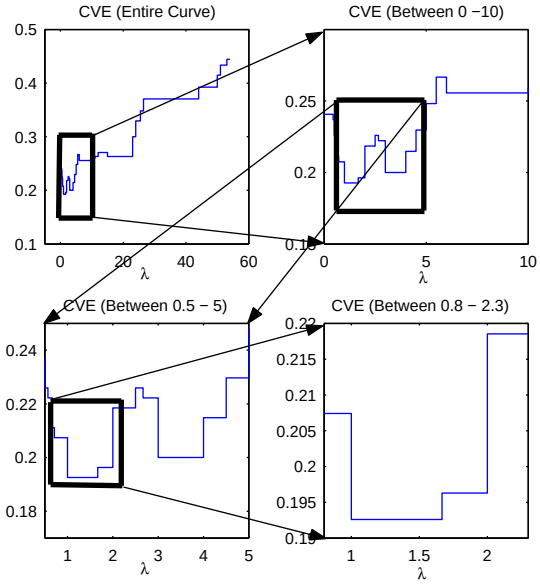

Figure 13: Using FBP in CV. The minimum of the CVE is found by zooming in.

2.5.1 Numerical Analysis of the Stability of the Cross-Validation Method

As mentioned earlier, we can use FBP to study (and graphically illustrate) the stability of CV. Here, we study a 10-fold CV. In each experiment, the data set is randomly and equally partitioned into ten subsets. The CV is to leave out one subset at a time, and uses the remaining nine subsets to train a model, then the omitted one is used to test (compute the testing error rate). This process is repeated for each subset, and the overall error rate is the average of the ten that are generated. Apparently in a 10-fold CV, the partition of the data set will change the result of CV. If CV is stable, the variation that is introduced by a different partition should be small. Is this always the case? By using the FBP, one can develop graphical tools to examine this condition.

We study the stability of the CV approach in the Wisconsin Breast Cancer data

(Blake and Merz, 1998). In Figure 14, five CV curves are plotted, showing that the optimal intervals for the penalization parameter λ are roughly in the same neighborhood.

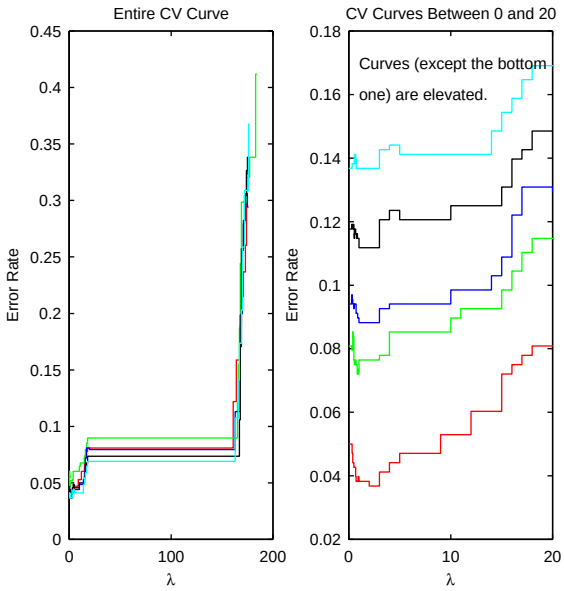

Figure 14: Five experiments of CV with the Wisconsin breast-cancer data. The left panel shows the entire CV curves. The right one focuses on the region between $(0, 20)$, the interval that includes the minima in all five experiments.

In Figure 15, 100 optimal intervals derived based on the CV principle are shown. Each row gives the location of the interval, inside of which when the parameter λ takes a value, the CVE is minimized. We will call these intervals *optimal intervals*. The figure shows the locations of these intervals out of 100 simulations. It demonstrates that even though in many cases the optimal intervals overlap significantly, in some cases an optimal interval is dramatically different from most of others. This demonstrates one application of FBP. Note that in Figure 15, the horizontal axis is logarithmic. More quantitative analysis on this phenomenon would be an interesting future research project.

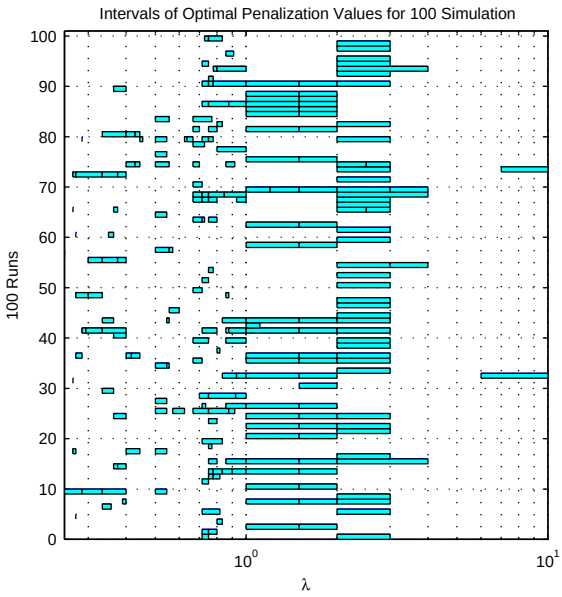

Figure 15: The locations of optimal intervals.

2.5.2 Difference Between the CV in CCP and the CV in FBP

A careful comparison between the CV that is described in section 11.5 of Breiman *et al.* (1984) and the CV in FBP, described at the beginning of this section reveals a significant difference between the two. We argue that the proposed method (i.e., CV in FBP) more faithfully realizes the principle of CV, while the CV in CCP examines only a subset of potential cases. An illustration of a case when the difference occurs will be provided at the end of this section.

To explain the difference, let us recall the CV procedure in CCP. More details are available in Breiman *et al.* (1984), so they are omitted. In CCP, the real axis for the penalization parameter λ is partitioned into intervals with boundaries:

$$\alpha_0 = 0 < \alpha_1 < \alpha_2 < \alpha_3 < \cdots < \alpha_K < +\infty = \alpha_{K+1},$$

where the constant K is equal to the size of an unpruned tree, and the above boundaries are computed based on pruning a tree with the entire data — equivalent to running FBP on the entire data set. In CCP plus CV, a special set of quantities are

chosen:

$$\{\alpha_j^* : \alpha_j^* = \sqrt{\alpha_j \alpha_{j+1}}, j = 1, 2, \ldots, K-1\}.$$

For a given λ, let CVE(λ) denote the cross validation error that was defined at the beginning of this section. The CV in CCP basically solves $\min_j CVE(\alpha_j^*)$, where $j \in \{1, 2, \ldots, K-1\}$ plus two cases corresponding to the two boundary intervals.

Apparently, the CV in CCP is computed based on a finite set of possible values of λ. This is not the CV principle that was described at the beginning of this section. FBP allows us to examine CVE(λ) for all possible value of λ. Based on this, we think that CV in FBP is more "faithful". Simulations will indicate that such faithfulness can be linked to better performance in applications. Figure 16 illustrates the difference between the two CV procedures, and when different optimal results will occur.

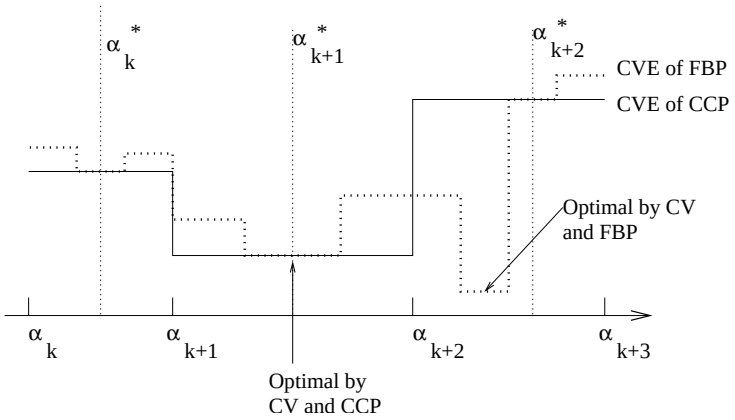

Figure 16: Difference between the two CV procedures (in CCP and FBP).

Our CV implementation of FBP is described here. For each leave-out set, we can compute the error rate (as e_{-i} in the previous description) as a piecewise-constant function *on the entire real line* of λ. In 10-fold CV, ten functions like this will be generated. The (point-wise) average of them is our CVE(λ), which provably is another piecewise-constant function. The optimal value of λ is the minimizer of this function. Figure 16 gives a graphical illustration of the difference of the CV procedure

Table 3: Iris example: λ's and error rates from 10-fold CV

λ	Error Rate
0	0.060
0.1667	0.0667
0.3333	0.0600
0.3571	0.0667
0.4000	0.0533
0.4167	0.0467
0.4545	0.0533
1.0000	0.0733
2.0000	0.0867
6.3333	0.1200
36.0000	0.1733
37.0000	0.2133
38.0000	0.2933
39.0000	0.3200
40.0000	0.3933
41.0000	0.4133
43.0000	0.5067
44.0000	0.6267
45.0000	0.6600
46.0000	0.7667

between CCP and FBP. The vertical dotted lines indicate where the CV-CCP tries to find the optimal values; the solid piecewise constant curve reveals the function that was minimized. The dotted curve indicates the function that was minimized by the CV-FBP approach. One can see that CV-CCP only consider an incomprehensive subset.

To illustrate the difference between CV-FBP and CV-CCP more clearly, we apply them to the Iris data (Blake and Merz, 1998). 10-fold CV is considered. Each iteration provides a set of λ's and corresponding error rates. Note that the error rate is a step function. Ten step functions are then averaged. Table 3 provides the average error rates for each interval of λ.

Now we explain the difference between CV-CCP and CV-FBP.

- CV-FBP considers all range of λ's (in Table 3) and finds the one that has the minimum CV error. In our example, the minimum CV error is 0.0467 and the

optimal λ is between 0.4167 and 0.4545.

- CV-CCP first uses the entire training data to find the partition of the λ-axis. Then the geometric means of pairs of adjacent λ's are computed. Table 4 shows the range of λ's and their geometric means. CV-CCP then chooses the optimal λ among the α^*'s. In our example, the minimum CV error in CCP is 0.0533 and the optimal λ is 0.4802. FBP produces smaller CV error than does CCP (0.0467 vs. 0.0533). In general, CV-FBP should always generate a smaller CV error than CV-CCP.

Table 4: Iris example: the range of λ's from the entire training-data set and geometric means

Tree size	13	7	5	3	2	1
α (range of λ)	0	0.3333	0.5000	1.5000	44.0000	50.0000
α^* (geometric means)	0	0.4802	0.8660	8.1240	46.9042	50.0000

In this example, both CV-FBP and CV-CCP lead to the same trained model: because the optimal λ from both methods are in the same interval $(0.3333, 0.5000]$, hence an identical tree is obtained. This leads to the same testing error rates for both methods. However, in many of our simulations, CV-FBP and CV-CCP lead to different testing errors. This can be seen in Tables 7 and 8 in the next section.

We have not addressed the problem of whether smaller CV errors lead to smaller testing errors. We leave it for future research.

2.6. Simulations

This section contains the following:

- Section 2.6.1 gives the simulation results that are related to the *inadmissible* tree sizes.

- Section 2.6.2 compares the CVE errors (a sanity check).
- Section 2.6.3 compares CCP and FBP for testing errors.
- Section 2.6.4 compares tree sizes.
- Section 2.6.5 illustrates the overall comparison.

The CCP algorithm is available from Salford Systems (2004) and has also been implemented by the authors in MATLAB. The following experimental setups are used throughout:

1. The Gini index is used as the impurity measure for tree growing.
2. 12 data sets available on the UCI database (Blake and Merz, 1998) are used. A detailed description of these data sets can be found in Appendix A.
3. We perform 10-fold cross validation. Each data set is split into two parts – a training set and testing set – ten times. At each time, 10-fold CVs are applied to the training set; the CVE, testing error, and tree size of the trained model are recorded. The reported values in the following tables are the averages of these ten simulations.

2.6.1 Inadmissibility

In all cases, the number of admissible tree sizes is significantly smaller than the total number of possible tree sizes. Here the tree sizes are measured by the number of terminal nodes i.e. in RDP they are the number of regions. The simulation results are presented in Table 5. An effective tree size is a tree size that is admissible. The *Effective* column gives the number of admissible tree sizes. The last column contains the number of terminal nodes in the un-pruned trees. In general, the number of admissible tree sizes is roughly 10% of all the possible sizes of the subtrees. As

mentioned in Section 2.1, we have this phenomenon mentioned loosely, but we have not seen a quantitative illustration of this fact.

Table 5: Comparison of the effective tree sizes with the sizes of the largest possible trees

Data Set	Effective	The Largest Tree Size
Australian Credit Approval	21	481
Cleveland Heart Disease	11	100
Congressional Voting Records	8	44
Wisconsin Breast Cancer	11	45
Iris Plants	6	13
BUPA Liver Disorder	13	132
PIMA Indian Diabetes	20	260
Image Segmentation	30	242
German Credit	26	344
Vehicle Silhouette	30	269
Waveform	20	367
Satellite Image	46	745

2.6.2 Cross-Validation Errors

Based on the explanation in Section 2.5.2 between CCP and FBP, one would expect that FBP *always* gives smaller CV errors. The simulations verify this, in Table 6. Statistical analysis was implemented as well. Comparing CCP with FBP, the p-value of the paired t-test is roughly 0.001. A 95% confidence interval for the mean difference of the CV error between CCP and FBP is $(0.0770, 0.2196)$. As mentioned earlier, we treat this as a "sanity check".

2.6.3 Comparison of Testing Errors

In simulation studies, testing errors are important. Table 7 gives the *average* testing errors for ten simulations of CCP and FBP (see more explanation at the beginning of this section). FBP tends to give smaller testing errors: six cases of smaller than CCP, and two ties, out of 12 simulations. However, the difference is not dramatic.

Table 6: Comparison of the CV error rates between CCP and FBP

Data Set	CCP	FBP	Winner
Australian Credit Approval	14.13	14.01	FBP
Cleveland Heart Disease	21.15	20.89	FBP
Congressional Voting Records	4.16	4.12	FBP
Wisconsin Breast Cancer	4.56	4.47	FBP
Iris Plants	5.20	5.07	FBP
BUPA Liver Disorder	31.27	31.03	FBP
PIMA Indian Diabetes	24.37	23.97	FBP
Image Segmentation	3.84	3.83	FBP
German Credit	24.61	24.48	FBP
Vehicle Silhouette	28.02	27.90	FBP
Waveform	22.86	22.83	FBP
Satellite Image	12.67	12.65	FBP

The paired t-test reveals that the p-value is 0.152. A 95% confidence interval for the mean difference of the testing errors (between CCP and FBP) is $(-0.0496, 0.2813)$. Note that zero is inside this interval.

Table 7: Comparison of the testing error between CCP and FBP

Data Set	CCP	FBP	Winner
Australian Credit Approval	14.84	14.58	FBP
Cleveland Heart Disease	26.82	27.19	CCP
Congressional Voting Records	5.50	5.60	CCP
Wisconsin Breast Cancer	5.09	4.94	FBP
Iris Plants	7.73	7.33	FBP
BUPA Liver Disorder	35.20	34.74	FBP
PIMA Indian Diabetes	25.34	25.26	FBP
Image Segmentation	5.80	5.81	CCP
German Credit	27.60	27.06	FBP
Vehicle Silhouette	30.25	30.27	CCP
Waveform	27.47	27.47	FBP, CCP
Satellite Image	12.85	12.85	FBP, CCP

2.6.4 Tree Sizes

In tree models, the size of a tree indicates the complexity of the model. In Table 8, the average tree sizes are given for 12 data sets; each has ten repetitions, as explained earlier. We observe that FBP gives smaller tree sizes than does CCP. Statistical analysis for the difference between CCP and FBP is performed. The p-value for the paired t-test is 0.006. A 95% confidence interval for the mean difference of tree size between CCP and FBP is $(0.617, 2.950)$. Based on this, it appears that a combination of CV and FBP generates smaller trees. This is just an empirical result, so theoretical understanding is needed.

Table 8: Comparison of the tree sizes (number of all nodes) between CCP and FBP

Data Set	CCP	FBP	Winner
Australian Credit Approval	10.60	8.00	FBP
Cleveland Heart Disease	8.20	6.80	FBP
Congressional Voting Records	7.00	5.80	FBP
Wisconsin Breast Cancer	14.20	10.60	FBP
Iris Plants	7.20	6.60	FBP
BUPA Liver Disorder	16.60	13.80	FBP
PIMA Indian Diabetes	15.20	9.00	FBP
Image Segmentation	89.40	87.20	FBP
German Credit	29.20	29.20	FBP, CCP
Vehicle Silhouette	63.40	62.60	FBP
Waveform	69.00	69.00	FBP, CCP
Satellite Image	249.00	249.0	FBP, CCP

2.6.5 Overall Comparison

Figure 17 compares three methods (C4.5, CCP, and FBP), based on the testing errors and tree sizes. Here, we include C4.5, which is another major tree-building method (Hamilton 2004). Note that it is not appropriate to compare C4.5 directly with CCP or FBP. Because C4.5 uses a different tree-growing algorithm (i.e., ID3 (Mitchell, 1997)) and generates multi-way splits, it allows nodes to have more than two child

nodes, while CCP and FBP are binary trees. However, it is still interesting to see the difference of their performance on some universal measures: smaller testing errors and smaller tree sizes, which are ideal classifiers. Based on this, being lower and left in Figure 17 is desired. Comparing the tree methods, we can see that FBP is relatively better than the other two methods. Only the first seven data sets are plotted in the figure, for legibility (the same trends were observed for all data sets).

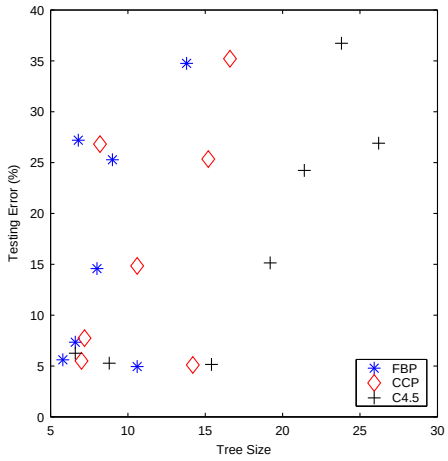

Figure 17: Testing errors vs. tree sizes: lower-left is optimal.

2.7. Structure of the FBP Algorithm

The algorithms are implemented in MATLAB. Cares is given to ensure the efficiency of each implemented algorithm. There are seven basic components:

1. *Tree Growing.* Grow a big and potentially redundant tree from data.

2. *Tree Pruning.* Use FBP to generate lists of linear functions at each node.

3. *Find Lower Bound.* Based on the list of linear functions at the root node, find the lower-bound function $f(\cdot)$.

4. *Identify the Best Subtree.* Given a value of the parameter λ, identify the size of the best subtree, together with the subtree itself.

5. *Testing.* Apply a tree model generated above to test data, and report the result.

6. *Application in CV.* Use FBP to realize CV.

Figure 18 provides an overview of the software architecture.

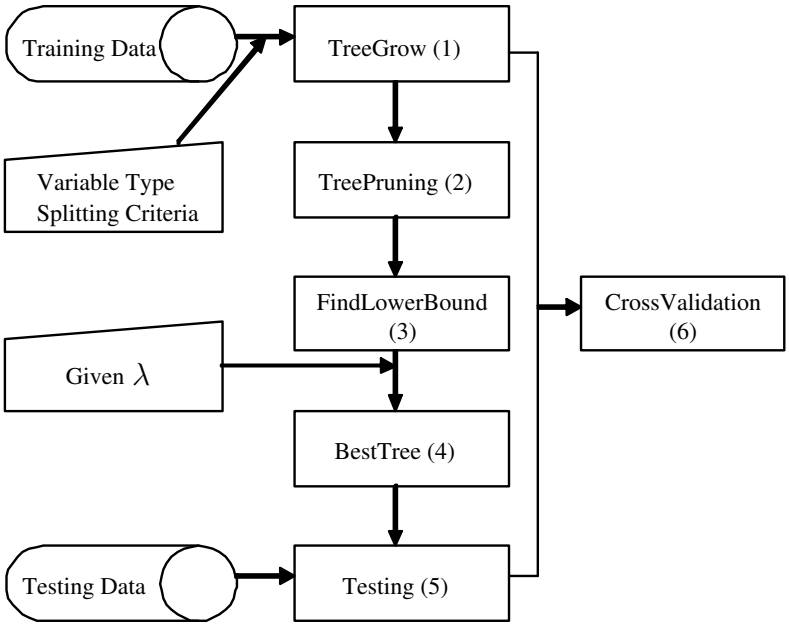

Figure 18: Relation between the functions. Numbers in the parentheses are steps.

2.8. Application: Gene Prediction

In this section we apply the FBP method to the problem of gene prediction. Gene prediction is to identify gene locations in sequentially identified DNA. Several biological terminologies are required and defined as follows:

1. *Gene:* A gene is the part of DNA, which controls hereditary information in an organism.

2. *Codon:* A codon is three consecutive nucleotides, which specifies one amino acid. There are 64 codons and 20 amino acids. Thus, the most of amino acids

are represented by more than one codon.

3. *Open Reading Frame (ORF):* ORF is a DNA sequence between start codon and stop codon. Usually start codons are ATG, TTG, or CTG and stop codons are TAA, TGA, or TAC, but they are dependent on an organism.

4. *Relative Synonymous Codon Usage (RSCU):* Number of times a particular codon is observed, relative to the number of times that the codon would be observed in the absence of any codon usage bias. RSCU can be calculated as follows:

$$RSCU_{xyz} = \frac{Obs_{xyz}}{Exp_{xyz}},$$

where Obs_{xzy} and Exp_{xyz} are the observed and expected frequencies of a codon xyz.

Table 10 contains the frequencies of synonymous codons of Alanine and the corresponding RSCUs from H. Pylori, one of the small genomes. For instance,

Table 9: RSCU for Alanine from H. Pylori

Amino acid	Codon	Number of codons	RSCU
A (Alanine)	GCA	5417	0.801
	GCC	5899	0.872
	GCG	6898	1.020
	GCT	8843	1.307
	Subtotal	27057	

RSCU of the codon, GCA, can be computed as follows: $RSCU_{GCA} = \frac{\frac{5417}{27057}}{\frac{1}{4}} = 0.801$. Moreover, the interpretation of RSCU can be made:

(a) if $RSCU_{xyz} = 1$, there is no codon usage bias

(b) if $RSCU_{xyz} < 1$, The codon xyz is used less frequently than expected.

(c) if $RSCU_{xyz} > 1$, The codon xyz is used more frequently than expected.

The main purpose of this study is to classify the ORFs into coding, coding potential, or noncoding using the compositional characteristics (e.g., frequency of nucleotide and dinucleotide, RSCU, and GC contents) and the length of the ORFs. We first extract all ORFs from a particular organism. We then assign each ORF to coding, coding potential, or noncoding based on a known result. An overview of problem is described as follows:

Input: An ORF sequence in a genome $S = (S_1, S_2, \cdots, S_k) \in \Omega, \Omega = A, C, G, T$.

Output: Labeling of sequence S whether it is a coding, coding potential, or noncoding region.

The classification results of three genomes (E.Coli, H. Pylori, and M. Genitalium) using the FBP method are described in Table 10.

Table 10: Classification accuracy of three genomes

Genome name	Genome length	Number of annotated genes	10-fold CV error
E.Coli	4,639,221	4,290	0.026
H.Pylori	1,667,877	1,717	0.057
M.Genitalium	580,073	476	0.049

In addition, we found that the important variables for the classification are length, the frequency of nucleotide, some of the frequency of dinucleotide and RSCU. For the H. Pylori case, important variables are as follows: Length, frequency of A,C,G, and T, frequency of AA, AG, CA, CC, CT, GA, GG, TC, TT, and RSCU (ATA, ATC, ATT, CTA, CTC, CTG, TTA, CTT, TTG, CCA, CCC, CCG, CCT, AGA, CGA, CGC, AGG, CGG, CGT, TCT, AGC, TCA, TCC, AGT, TCG, ACA, ACC, ACG, ACT, GTA, GTC, GTG, and GTT).

2.9. Conclusions

A FBP algorithm was proposed, which provides a graphical way to implement the task of minimizing CPLF. FBP has the same objective as does CCP; however it is more advantageous because it provides a full spectrum of information in tree pruning. It can be used to realize the principle of CV more "faithfully". A combination of FBP and CV render "better" classifiers in simulations, compared to other existing methods. Simulation results on real data sets render several other interesting findings; for example, the number of admissible tree sizes is always a small proportion (roughly 10%) of the number of all possible tree sizes. Computational -complexity analysis is provided. This method is appealing in implementation and has the potential to be used in other tree- related studies, e.g., the stability of applying CV in building tree models.

CHAPTER III

PERFORMANCE OF CROSS VALIDATION ON TREE-BASED MODELS

3.1. Introduction

Cross Validation (CV) was described as early as Stone (1974). It has been of tremendous interest to characterize why and how a CV method works. To our knowledge, most of the theoretical work on CV concentrates on *regression* applications rather than *classification.* Some well cited works include Efron (1983, 1986), Shao (1993, 1996, 1998), and Zhang (1992, 1993a, 1993b). Of special value is Zhang's description of a distributional property of CV for linear regression models. For the problems of model selection and error prediction in linear models, certain forms of CV are shown to be equivalent to well known model selection criteria such as the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and the C_p statistics. Based on this framework, good performance of CV and asymptotic convergence can be established.

In the regression problem, the risk function is continuous. Hence it is relatively easy to study the behavior of CV. However, in classification problems, nonlinearity related to categorical response makes it hard to establish an equivalence between CV and some existing criteria. Despite some contributions in this direction, e.g., Lachenbruch and Mickey (1968), most of the theoretical questions remain open.

In this Chapter, an experimental approach is introduced to illustrate the behavior of the tree-based classifier selected by the CV method (called the *"cross-validated tree classifier"*). The following is a synopsis of the approach.

1. *Oracle classifier.* Given the distribution of point clouds derived from likelihood ratio of the Neyman-Pearson, an optimal classification rule is derived. Since one needs to know the underlying distribution, such a classifier is called the *oracle classifier.*

2. *Cross-validated classifier.* Given a training set, a classifier can be trained by a minimized average error rate given in the form of CV. The description of the CV error rate is presented in the following sections. This classifier is called the *cross-validated classifier.*

3. *Training and testing errors.* Both of the above classifiers can be applied to the training and testing sets. In general, a smaller error rate on the training set does *not* necessarily mean optimality, because it may be introduced by overfitting. For a classifier, equality between training error and testing error may be desirable. Moreover, if the oracle classifier is applied to both training and testing sets, the difference between the two error rates should be small since the difference is only affected by sampling error.

$$(\text{testing error - training error})_{oracle} \approx 0$$

   On the other hand, if the testing-to-training error difference is huge, the randomly sampled data does *not* reflect the underlying distribution. This suggests that the classifier selected is inappropriate.

4. *Methodology evaluation.* Based on previous analysis, the following method can be used to analyze a cross-validated classifier. The difference between the training error and the testing error is calculated for the cross-validated classifier. Let $e_{1,A}$ and $e_{2,A}$ respectively denote the *training* error of the oracle and the cross-validated classifiers, where

- "1" stands for oracle classifiers,
- "2" stands for cross-validated classifiers, and
- "A" stands for the training set.

Let $e_{1,B}$ and $e_{2,B}$ denote the two corresponding *testing* error rates, where

- "B" stands for testing set.

We consider the differences:

$$e_{2,B} - e_{2,A} \qquad \text{vs.} \qquad e_{1,B} - e_{1,A}.$$

5. *Main observation.* The main observation is that the above two quantities have a roughly statistically linear relationship. This is more evident in Figure 23. Let $D_1 = e_{1,B} - e_{1,A}$ and $D_2 = e_{2,B} - e_{2,A}$, we have

$$D_1 = C \cdot D_2 + \varepsilon, \tag{7}$$

where the constant C, $|C| \leq 1$, depends on the underlying distribution, and the random variable ε has zero mean and seemingly normal distribution.

In our simulations, the data are generated according to known distributions. Based on the distributions, two classifiers are considered: the oracle classifier and the cross-validated tree classifier. In most cases, we observe the phenomenon that is depicted in (7). The influence of the decision-boundary geometry, the parameter of the underlying distributions, and the sample size training are studied in the simulations.

The above study became feasible due to a new algorithm – frontier-based tree-pruning algorithm (FBP) by Huo et al. (2004). The FBP allows the implementation of CV in a tree model which yields satisfactory accuracy and efficiency.

3.2. The Cross-Validation Principle

Suppose we have two disjoint sets: a training and a testing set. The former set is used to learn the model and the latter to evaluate the performance of the trained model. The framework of a generic validation process is illustrated in Figure 19 and summarized in 5 steps.

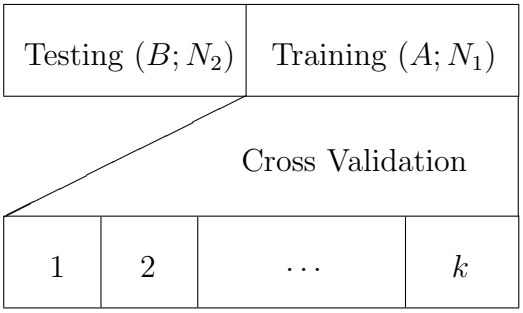

Figure 19: A structure of the CV process.

Step 1. Divide the data into training, and testing sets.

Step 2. Train the model using the training set.

Step 3. Select the parameter(s) of the model using the training set via CV.

Step 4. Select the best model from steps 2 and 3.

Step 5. Assess the final model using the testing set.

Let A denote the training set of size N_1 and B the testing set of size N_2. Let F denote the common underlying rule for both sets. We consider a k-fold CV and α is an algorithmic parameter of a model. If we denote $e_\alpha^{(-i)}$ as the error rate when excluding the ith folder during CV, the cross-validating error at α is given by

$$CV(A; \alpha) = \frac{1}{K} \sum_{i=1}^{k} e_\alpha^{(-i)}$$

The principle of CV is to choose an α such that $CV(A;\alpha)$ is minimized:

$$\alpha_0 = \underset{\alpha}{\operatorname{argmin}}\, CV(A;\alpha)$$

Let $T_{\alpha_0}(A)$ denote the model that is built by using $\alpha = \alpha_0$ and the training sample A. We then have two different errors: training error based on CV and testing error. The former can be expressed as $e_{CV}(A;\alpha_0)$, which the model $T_{\alpha_0}(A)$ produces. The testing error can be represented as $e_T(T_{\alpha_0}(A), B)$, which denotes the error rate when the model $T_{\alpha_0}(A)$ is applied to the data B. The quantities described here can be summarized as follows:

$$\begin{aligned}
A &\Rightarrow \alpha_0 = \underset{\alpha}{\operatorname{argmin}} \frac{1}{K}\sum_{i=1}^{K} e_{\alpha}^{(-i)} = \underset{\alpha}{\operatorname{argmin}}\, CV(A;\alpha) && \text{CV} \\
&\Rightarrow T_{\alpha_0}(A) && \text{optimal model} \\
&\Rightarrow e_{CV}(A;\alpha_0) && \text{training error based on CV} \\
A, B &\Rightarrow e_T(T_{\alpha_0}(A), B) && \text{testing error}
\end{aligned}$$

3.3. Cross Validation in a Tree-Based Model

3.3.1 Frontier-Based Tree-Pruning Algorithm

Huo *et al.* (2004) proposed a frontier-based tree-pruning (FBP) method which provides the full spectrum of information regarding tree pruning, such as (1) given the value of the penalization parameter λ, this method gives the minimum size of a decision tree; (2) given the size of a decision tree, it provides the range of the penalization parameter λ, in which the cost penalization approach will render a tree that has the same size; and (3) this algorithm can tell the sizes of trees that will be definitely inadmissible—no matter what the value of the penalty parameter is, the resulting tree of a complexity-penalization approach will not have that size. Moreover, this method showed that a combination of the CV and the FBP would facilitate a reduction in the testing errors. The main idea of this method considers complexity-penalized loss

function (CPLF), which is defined in equation (8) and searches the entire set of a penalizing parameter, denoted by α to find the optimal tree using CV. Let $L(T_b)$ denote the loss function associated with tree T_b and $|T_b|$ denote the size of the tree T_b. The associated complexity-penalized loss function is

$$L(T_b) + \alpha|T_b|, \tag{8}$$

where α is a penalizing parameter. The principle of minimizing CPLF is to choose a subtree T_b that minimizes the CPLF. In other words, the principle of minimizing CPLF is to solve

$$T_{b_0} = \underset{T_b}{\operatorname{argmin}} \quad L(T_b) + \alpha|T_b|. \tag{9}$$

3.3.2 Cross Validation with FBP

Suppose the observations are

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_N, y_N)\},$$

where N is the number of observations, x_i's are predictor variables, and y_i's are responses. Note that x_i's can be multivariate. Suppose the above set is (roughly equally) partitioned into k subsets:

$$S_1 \cup S_2 \cup \cdots \cup S_k.$$

At each time, if we leave out one subset (say S_i) and use the remaining sets to grow a tree and then prune a tree, the lower bound function will be denoted as $f_{-i}(\alpha)$, where $f(\alpha)$ is the minimum value of the CPLF in (8). For each value of the parameter α, the size of the optimal subtree and the subtree itself can be extracted. The optimal subtree determines a model which is then applied to the left-out subset (which is S_i). This is equivalent to *testing*. The error rate in testing can be computed and is denoted by $e_{-i}(\alpha)$. Note that functions $f_{-i}(\alpha)$ and $e_{-i}(\alpha)$ are of the same variable.

Because function $f_{-i}(\alpha)$ is a piecewise linear function, it is not hard to prove that function $e_{-i}(\alpha)$ is also a piecewise constant function (or step function). The principle of CV is to find the value of the parameter α that the value of the average of e_{-i}'s

$$\frac{1}{K}\sum_{i=1}^{K} e_{-i}(\alpha)$$

is minimized. This principle can be easily implemented by using the FBP method. The generation of functions $f_{-i}(\alpha)$ is already in the FBP. The generation of functions $e_{-i}(\alpha)$ can be easily implemented. For more details, readers can refer to Huo *et al.* (2004). Figure 20 illustrates the trend of $e_{CV}(A;\alpha)$ against α. By this we mean that the error rates vary depending on α. The lowest part of the step function indicates the optimal α ($=\alpha_0$).

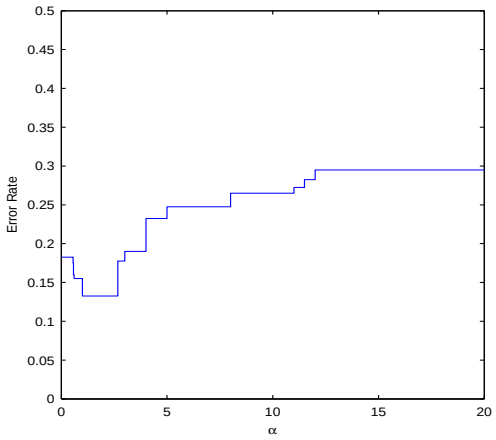

Figure 20: The range of the optimal α that produces the smallest error rate.

3.4. Analysis

In this section, we describe some distributional analysis. Suppose the data is again divided into a training set and a testing set. If there is an oracle, who knows the underlying distribution, he/she can derive a classifier, which works statistically optimal – having the minimum testing error overall, following the principle of Neymann-Pearson. We call such a classifier an Oracle Classifier (OC). Note that this classifier

does not depend on the sampled data. Let $e_{1,*}$ denote the error rate by applying the OC to data $*$. Then we can obtain $e_{1,A}$ and $e_{1,B}$ using the following equations:

$$e_{1,A} = \frac{1}{N_A}\sum_{Y_A} I(\hat{Y}_{OC(X_A)}, Y_A), \tag{10}$$

$$e_{1,B} = \frac{1}{N_B}\sum_{Y_B} I(\hat{Y}_{OC(X_B)}, Y_B), \tag{11}$$

where N_* is the size of data $*$ and I is a 0-1 loss function defined as follow:

$$I(\hat{Y}(X), Y) = \begin{cases} 0, & \text{if } Y = \hat{Y}(X), \\ 1, & \text{if } Y \neq \hat{Y}(X). \end{cases}$$

Also, we can compute their difference:

$$D_1 = e_{1,B} - e_{1,A}. \tag{12}$$

Proposition 3.4.1 *When errors $e_{1,A}$ and $e_{1,B}$ are defined as in equations (10) and (11), we have $e_{1,A} \sim \mathcal{N}(p, \sigma^2_{e_{1,A}})$ and $e_{1,B} \sim \mathcal{N}(p, \sigma^2_{e_{1,B}})$ where p is the true risk. Therefore, $D_1 = e_{1,B} - e_{1,A} \sim \mathcal{N}(0, \sigma^2_{D_1})$.*

Proof. $I(\hat{Y}_{OC(X_A)}, Y_A)$ can be described as an independent and identical Bernoulli distribution with p, where p is the true risk. Independency and identity are hold since the decision boundaries of an oracle classifier are fixed in each experiment. Therefore, $\sum_{Y_A} I(\hat{Y}_{OC(X_A)}, Y_A)$ follows a Binomial distribution with N and p, where N is the number of experiments. This can be approximated by $\mathcal{N}(Np, Np(1-p))$. Thus, $e_{1,A}$ can be described as $\mathcal{N}\left(p, \left(\sqrt{\frac{p(1-p)}{N_A}}\right)^2\right)$. Similarly, we have $e_{1,B} \sim \mathcal{N}\left(p, \left(\sqrt{\frac{p(1-p)}{N_B}}\right)^2\right)$. Furthermore, because the difference of two Normal distributions is also a Normal distribution, $D_1 = e_{1,B} - e_{1,A}$ will also follow a approximate Normal distribution with mean 0 and variance equals to $\frac{p(1-p)}{N_A} + \frac{p(1-p)}{N_B}$. □

Note that depending on the sampled data, D_1 is not necessarily zero. However, since it only depends on the sampling errors, its expectation should be around zero and its variance ($\sigma^2_{D_1}$) is in some sense the minimum. The following is a conjecture.

Conjecture 3.4.2 *Suppose D_ξ is the difference between testing and training errors in any other classifier. We have $\sigma^2_{D_1} \leq \sigma^2_{D_\xi}$.*

In the following paragraph, we briefly review some known general principles. The proof of the above conjecture can be derived by following general principles, with more technical detail.

The target space (T) can be defined as the space of the functions, containing the ideal classifier that minimize the risk. And the hypothesis space (H) can be defined as the space of functions that a learning algorithm is allowed to search. Several risks (from T and H) can be defined as follows:

- E_{f_T}: The true risk of the best function in T,
- E_{f_H}: The true risk of the best function in H, and
- E_{f_S}: The empirical risk of the function in H we actually find.

Sampling error, which depicts the difference between the best function in H and the function in H we actually find can be represented as $SE = E_{f_S} - E_{f_H}$. The sampling errors occur because our finite sample does not give us enough information to choose the best function in H. Approximation error is the difference between the true risk in H and T, which can be represented as $AE = E_{f_H} - E_{f_T}$. This error occurs because H is smaller than T. Based on the relations described above, we can formulate our empirical risk as the sum of sampling error, approximation error, and the true risk in T: $E_{f_S} = SE + AE + E_{f_T}$. In an oracle classifier, AE is always "in some sense" equal to zero because the target space and the hypothesis space are the same. In any other classifier, however, AE can be greater than or equal to zero. Hence, the variance of the error difference in an oracle classifier should always be less than the corresponding variance in any other classifier.

Incorporating CV in a tree-based model yields a classifier, called a cross-validated tree classifier (CVT). Let $e_{2,A}$ denote the training error based on CV, which is the error rate by applying CVT to the training data (equation (13)). Let $e_{2,B}$ denote the testing error when the CVT is applied to the testing data (Equation (14)).

$$e_{2,A} = \frac{1}{N} \sum_{Y_A} I(\hat{Y}_{CVT(X_A)}, Y_A); \tag{13}$$

$$e_{2,B} = \frac{1}{N} \sum_{Y_B} I(\hat{Y}_{CVT(X_A)}, Y_B). \tag{14}$$

A quantity similar to the one in Equation (12) is

$$D_2 = e_{2,B} - e_{2,A}. \tag{15}$$

One can still argue that the variance D_2 can be decomposed into two components which are sampling and approximate errors described above. We have the following conjecture.

Conjecture 3.4.3 *For error difference D_2 that is described in (15), we have*

$$D_2 \sim \mathcal{N}(0, \sigma^2_{D_2}).$$

The distribution of $e_{2,A}$, $e_{2,B}$, and D_2 cannot be derived directly because of the correlation between iterations of a CV procedure. Since distributions of each iteration in CV (e.g., 10 iterations in 10-fold CV) are correlated: one can not simply apply asymptotic approximation in this case. Instead of a theoretical proof, we analyze their distributions empirically.

Figure 21 illustrates the normal probability plots of three types of errors, i.e., $e_{2,A}$, $e_{2,B}$, and D_2. The errors are generated under the rectangular decision boundary with parameter $p = 0.1$ and sample size 800: training 400, testing 400 (more details are provided in the next section). It suggests that all three quantities follows Normal distributions with corresponding means and variances.

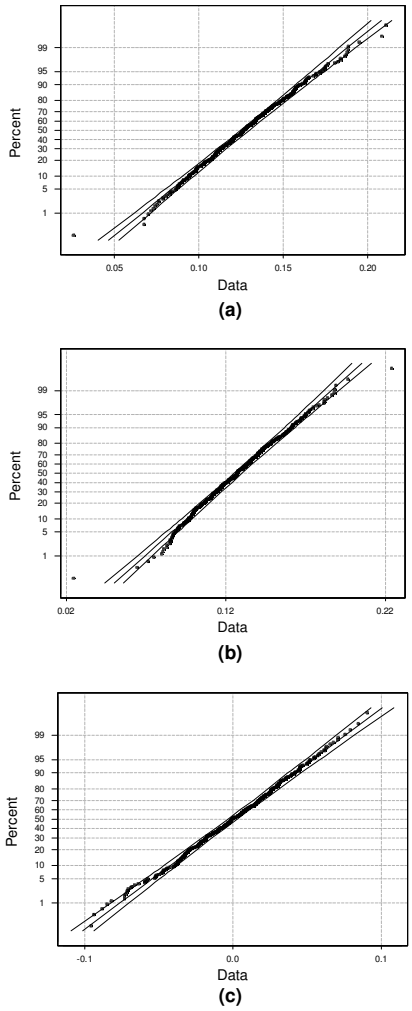

Figure 21: Normal probability plot for the errors in a cross-validated tree classifier (CVT). (a) training error for the CVT $(e_{2,A}) \sim \mathcal{N}(0.127, 0.03)$, (b) testing error for the CVT $(e_{2,B}) \sim \mathcal{N}(0.127, 0.03^2)$, and (c) error for the difference $(D_2 = e_{2,B} - e_{2,A})$ $\sim \mathcal{N}(0.000, 0.03^2)$.

Now we consider a statistical relation between D_1 and D_2, using a linear regression method. Our main conjecture is the following.

Conjecture 3.4.4 *Given the error differences that are defined in (12) and (15), we have*

$$D_1 = C \cdot D_2 + \varepsilon, \tag{16}$$

where C is a constant and its range should be between 0 and 1, and ε is a random error satisfying $\mathcal{N}(0, \sigma^2)$.

In this chapter we perform simulations to justify the conjectures. The interpretation of such a result is that the equality of errors between testing and training set from a CVT is comparable with that of an OC up to a constant C.

3.5. Simulations

3.5.1 Setup

We consider three different decision boundaries (denoted by $\mathcal{B}$) inside a unit square (denoted by $\mathcal{S}$) and a underlying rule F.

Decision Boundaries, $\mathcal{B}$

- *Case 1:* X $\in \mathcal{B}$ where $\mathcal{B}$ is a rectangular decision boundary, which is $0.2 \leq X_1 \leq 0.8$ and $0.3 \leq X_2 \leq 0.8$.
- *Case 2:* X $\in \mathcal{B}$ where $\mathcal{B}$ is a circular decision boundary, which is $(x_1 - 0.5)^2 + (X_2 - 0.5)^2 < 0.2^2$.
- *Case 3:* X $\in \mathcal{B}$ where $\mathcal{B}$ is a triangular decision boundary, which is $X_2 > 0.2$, $X_2 < 2X_1 - 0.2$, and $x_2 < -2X_1 + 1.8$.

Rule, $F(\mathcal{B}, p)$

- If input $X \in \mathcal{B}$, then response

$$Y = \begin{cases} 1, & \text{with probability (w.p.) } p, \\ 0, & \text{w.p. } 1 - p; \end{cases}$$

- If input X $\notin \mathcal{B}$, then response

$$Y = \begin{cases} 1, & \text{w.p. } p - 1, \\ 0, & \text{w.p. } p. \end{cases}$$

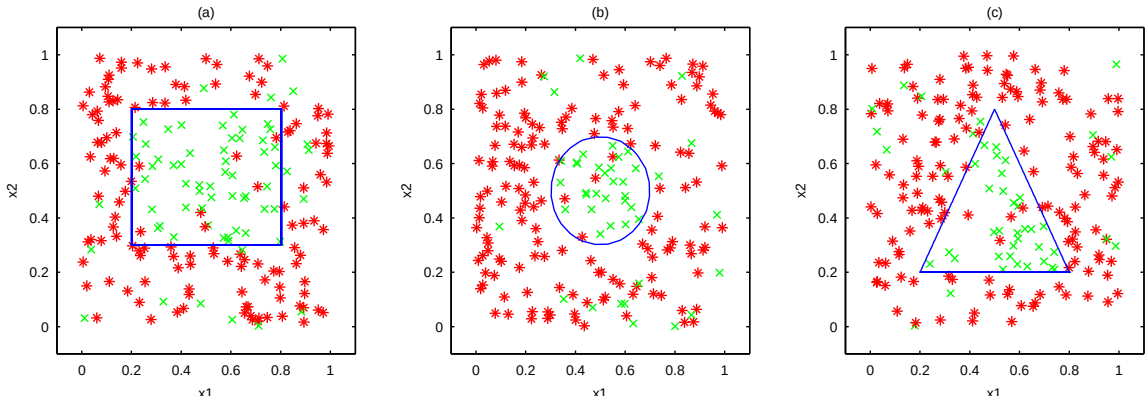

Figure 22: Illustration of 200 simulated data sets with three different decision boundaries. (a) Rectangular decision boundary, (b) Circular decision boundary, and (c) Triangular decision boundary.

Figure 22 illustrates the three different decision boundaries and data points generated by the underlying rule. Two hundred simulated data (training:100, testing:100) with $p = 0.1$ is utilized. For each randomly generated data, based on a underlying rule F, we can obtain a series of error rates ($e_{1,A}$, $e_{2,A}$, $e_{1,B}$, $e_{2,B}$), defined in previous section. Note that we employ 10-fold CV to compute $e_{2,A}$ and $e_{2,B}$, because many studies showed that 10-fold CV produced decent results. (Zhang, 1992; Brieman and Spector, 1989).

3.5.2 Relation Between D_1 and D_2

To identify a statistical relation between D_1 and D_2, we utilize linear regression analysis. D_1 is taken as the response variable and D_2 as the predictor variable.

Figure 23 represents linear regression between D_1 and D_2 with three different decision boundaries. We consider $p = 0.1$ and the 200 sample size. It suggests that there is a statistical relation between two variables. Slopes and intercepts of regression lines are shown in Tables 11 and 12. Note that the number of experiments do not significantly affect the slope. Moreover, Table 12 shows that intercepts are not significant in most cases. Thus, each case of the regression function can be presented

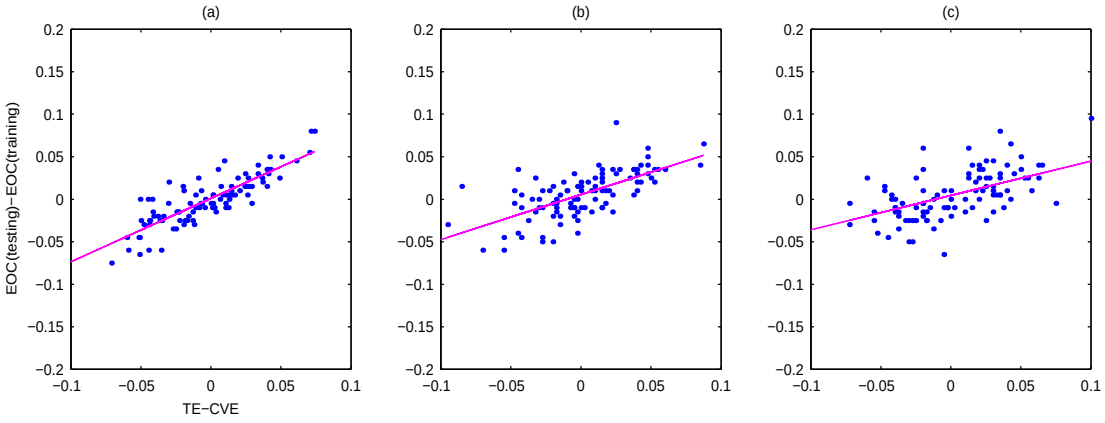

Figure 23: Regression plots between D_2 and D_1. (a) Rectangular decision boundary, (b) Circular decision boundary, and (c) Triangular decision boundary.

as follows using the average of the slopes (indicated in the 9th column of Table 11),

$$\begin{aligned} E_\alpha\{D_1\} &= 0.766 \cdot D_2, \\ E_\beta\{D_1\} &= 0.511 \cdot D_2, \\ E_\gamma\{D_1\} &= 0.459 \cdot D_2, \end{aligned}$$

where α, β, and γ respectively represent rectangular, circular, and triangular decision boundaries. The slope parameter 0.766 in a rectangular decision boundary indicates that the expected difference between testing and training error from an oracle classifier is 0.766 times that of a cross-validated tree classifier. Similar interpretations can be made with respect to the circular and triangular decision boundaries.

3.5.3 Effects of the Geometry of Decision Boundaries

Figure 24 shows that the relationship of the difference between testing and training error of the OC and CVT is affected by the geometry of the decision boundaries. The larger value of the slope, the less difference between D_1 and D_2. It is not hard to imagine why a rectangular decision boundary has a larger value of the slope than other boundaries. This is due to the characteristic of the recursively binary splitting of the feature space in tree-based methods. Furthermore, Table 13 shows the R^2 (coefficient

Table 11: Slopes in a regression line with differently shaped decision boundaries and number of experiments. The values in the parentheses indicate the slopes in a regression line through the origin

# of exp.	20	50	100	200	300	400	500	Mean	Stdev
Rectangle	0.852	0.635	0.745	0.764	0.747	0.797	0.775	0.759	0.066
	(0.852)	(0.703)	(0.741)	(0.760)	(0.734)	(0.797)	(0.774)	(0.766)	(0.048)
Circle	0.501	0.529	0.528	0.502	0.516	0.548	0.528	0.522	0.017
	(0.476)	(0.518)	(0.529)	(0.494)	(0.511)	(0.531)	(0.519)	(0.511)	(0.020)
Triangle	0.525	0.489	0.403	0.459	0.485	0.387	0.516	0.466	0.054
	(0.530)	(0.433)	(0.409)	(0.460)	(0.483)	(0.388)	(0.513)	(0.459)	(0.053)

Table 12: Intercepts in regression lines and their significance with different decision boundaries and the number of experiments. The values in the parentheses indicate the p-values of intercepts

# of exp.	20	50	100	200	300	400	500
Rectangle	0.00233	0.0094	0.00094	-0.00174	0.00121	-0.000586	0.00044
	(0.682)	(0.624)	(0.856)	(0.108)	(0.802)	(0.462)	(0.573)
Circle	0.00230	0.00218	0.00534	0.00084	0.00162	0.00420	0.00224
	(0.682)	(0.551)	(0.011)	(0.656)	(0.195)	(0)	(0.024)
Triangle	0.00705	0.00478	0.00446	-0.00009	0.00114	-0.00018	0.00108
	(0.164)	(0.254)	(0.064)	(0.959)	(0.410)	(0.534)	(0.644)

of determination) of each boundary. It also shows that the rectangular boundary has larger R^2 than the others. This result suggests that a strong degree of linear association between D_1 and D_2 exists within a rectangular decision boundary. In other words, a cross-validated tree classifier based on rectangular decision boundary behaves more like the oracle classifier than for the other geometries. Note that in Tables 11 and 13 the number of experiments do not significantly affect the slope and R^2.

3.5.4 The Effect of the Parameters in an Underlying Distribution

Recall that an underlying distribution in our simulation is Bernoulli and its parameter is the probability of any particular points being inside the decision boundary.

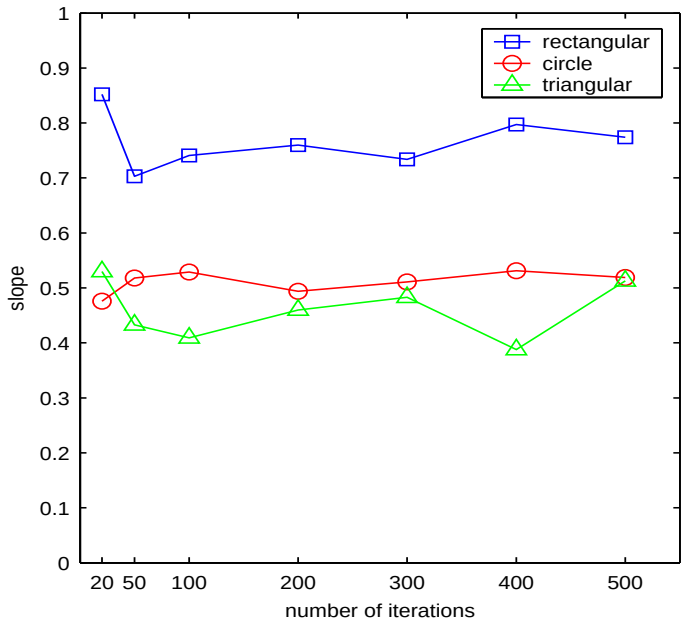

Figure 24: Slopes in a regression line with different decision boundaries and sample sizes.

Table 13: R^2 (Coefficient of Determination) with different decision boundaries and number of experiments

# of exp.	20	50	100	200	300	400	500	Mean	Stdev
Rectangle	0.878	0.705	0.733	0.718	0.643	0.729	0.674	0.725	0.074
Circle	0.339	0.441	0.459	0.430	0.456	0.431	0.467	0.432	0.043
Triangle	0.449	0.282	0.316	0.320	0.370	0.388	0.414	0.362	0.059

Table 14 describes the slopes of regression lines with different parameters based on a rectangular decision boundary. The other geometries of decision boundaries give similar results. Figure 26 is the box plot of the slopes in different parameter values. It shows that parameter values between 0.1 and 0.2 produce a strong linear relationship between the OC and the CVT but this relationship becomes weaker as the parameter value becomes either extremely small or close to 0.5. It's not difficult to explain why D_1 and D_2 have a weak linear relationship as p approaches to 0.5. If p equals 0.5, we have the same probability for each class being inside or outside of a decision

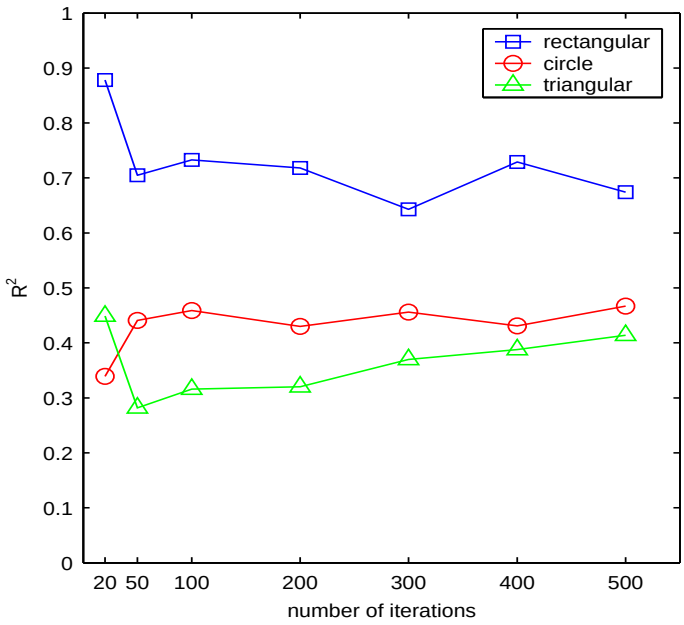

Figure 25: R^2 (Coefficient of Determination) in a regression line with different decision boundaries and number of experiments.

boundary. In this case, classification processes are mostly affected by random effects instead of the decision rule. This randomness causes a weak relationship between the two classifiers. For small p (e.g., p=0.01), the relationship of two classifiers is very sensitive to the changes of error rates because both classifiers produce very small error rates. This high sensitivity results in a relationship between the two classifiers that is relatively weak.

Table 15 and Figure 27 show the R^2 for the above regression analysis. These results show that when the slope is large, there is a stronger case for the existence of a regression model.

3.5.5 The Effect of the Sample Size

In this section, we study the relationship of the equality between testing and training errors from both classifiers with different sample sizes. First we consider five different sample sizes (training + training): $100, 200, 300, 400, 500$. For each sample size, we

Table 14: Slopes in a regression line with different parameters. The values in the parentheses are the slopes in a regression line through the origin

	0.01	0.05	0.1	0.2	0.3	0.4	0.5
1	0.261	0.631	0.747	0.747	0.700	0.506	0.120
	(0.261)	(0.626)	(0.741)	(0.741)	(0.700)	(0.522)	(0.099)
2	0.352	0.624	0.875	0.810	0.458	0.275	0.002
	(0.339)	(0.623)	(0.874)	(0.810)	(0.432)	(0.228)	(0.033)
3	0.272	0.460	0.743	0.790	0.714	0.496	-0.181
	(0.279)	(0.561)	(0.743)	(0.789)	(0.680)	(0.400)	(-0.162)
4	0.348	0.570	0.770	0.619	0.458	0.157	0.158
	(0.359)	(0.570)	(0.767)	(0.624)	(0.419)	(0.120)	(0.179)
5	0.301	0.690	0.714	0.822	0.542	0.481	-0.089
	(0.301)	(0.690)	(0.716)	(0.716)	(0.535)	(0.397)	(-0.101)
Average	0.307	0.595	0.770	0.758	0.575	0.082	
	(0.301)	(0.614)	(0.768)	(0.736)	(0.553)	(0.107)	
S.D.	0.042	0.087	0.062	0.083	0.126	0.080	
	(0.041)	(0.052)	(0.062)	0.073	(0.133)	(0.101)	

consider five different ratios of testing to the training samples: 1:3, 1:2, 1:1, 2:1, 3:1. Table 16 shows the slopes in a regression line from different ratios of the training and the testing sample sizes. Again, since the intercepts in a regression line are not statistically significant, we consider the slopes with zero intercept shown in the parentheses in Table 16. Figure 28 illustrates a three-dimensional contour plot. The X and Y-axes repectively represent the sample size and the ratio of testing to training samples. For instance, if the values on the X and Y-axes are 300 and 2, the experiment has a training sample size of 100 and 200 for the testing sample. The Z-axis (the values on the contour plot) indicates the slopes of each regression line. This plot provides a nice guideline for determining the ratio of testing to the training sample size for achieving the targeted performance. For instance, if we want our cross-validated tree classifier to be $\frac{1}{0.84677}$ of the oracle classifier, the corresponding values on the X-axis give the ratio corresponding to the different total sample sizes. In addition, we observe that the slopes change a lot when the size of sample is less than 300 but stabilize

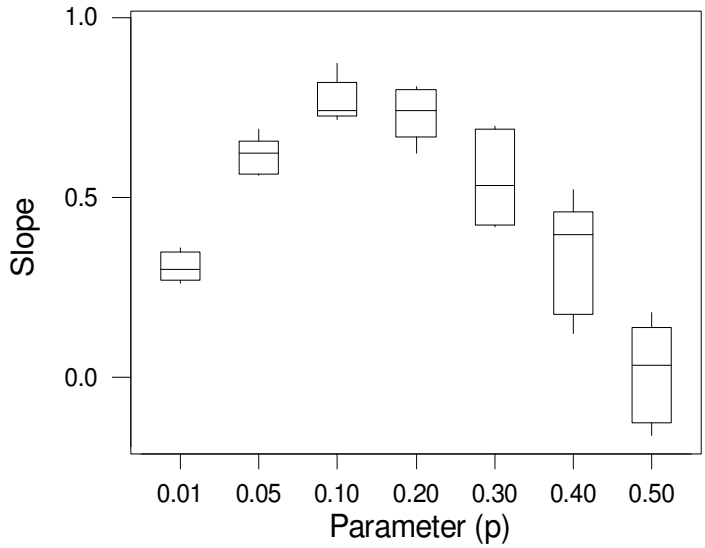

Figure 26: Slopes in a regression line with different parameters.

when sample size becomes larger than 300. This implies that in this example, the sample size 300 is sufficient for good performance of the tree classifier compared to the oracle classifiers with respect to testing and training error.

3.6. Conclusions

In this chapter, we present a simulation study that compares a cross-validated tree classifier with an oracle classifier based on the knowledge of an underlying distribution. The main contribution of this chapter is to experimentally explore the statistical relationship of the difference between testing and training errors from two classifiers via a linear regression model. Simulation results indicate that the intercept of the regression line is zero. These results suggest that the difference between testing and training errors from a cross-validated tree classifier is a constant factor of that of an oracle classifier, within a constant factor. Various simulations appear to justify the

Table 15: R^2 in a regression line with different parameters

	0.01	0.05	0.1	0.2	0.3	0.4	0.5
1	0.206	0.609	0.733	0.627	0.443	0.165	0.008
2	0.390	0.569	0.804	0.706	0.218	0.048	0.006
3	0.241	0.330	0.743	0.673	0.424	0.112	0.024
4	0.309	0.520	0.741	0.529	0.229	0.014	0.012
5	0.342	0.684	0.637	0.716	0.362	0.125	0.006
Average	0.307	0.595	0.770	0.758	0.575	0.093	0.011
S.D.	0.042	0.087	0.062	0.083	0.126	0.054	0.007

authors' conjectures. Regression slopes and R^2 are used to measure of the degree of the relationship. Both the slope and R^2 being equal to 1 suggest a strong relationship between two classifiers. Additionally, we demonstrate that the above relationship is influenced by other factors such as the geometry of the decision boundaries, the probabilistic parameter of an underlying distribution, and sample size.

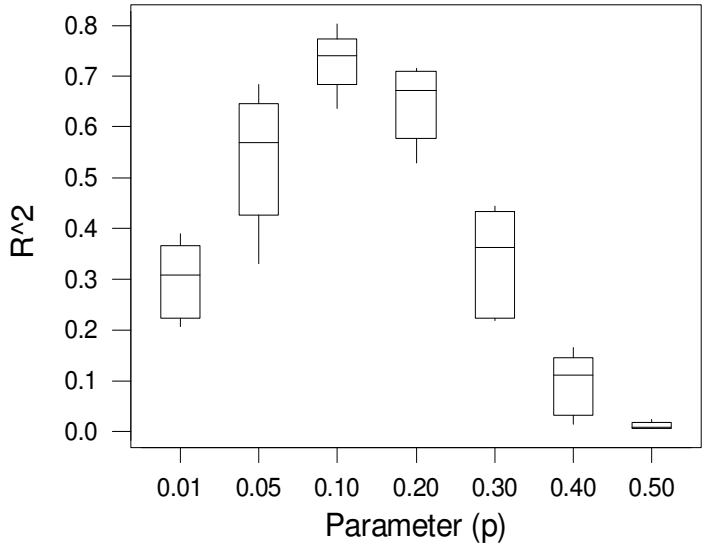

Figure 27: R^2 in a regression line with different parameters.

Table 16: Slopes in a regression line with different sizes and ratio of training and testing sets. The values in the parentheses indicate the slopes in a regression line through origin

	100	200	300	400	500	Average	S.D.
3:1	0.557	0.710	0.790	0.858	0.861	0.755	0.127
	(0.553)	(0.718)	(0.786)	(0.859)	(0.861)	(0.755)	(0.127)
2:1	0.401	0.696	0.740	0.859	0.906	0.720	0.198
	(0.407)	(0.700)	(0.740)	(0.864)	(0.906)	(0.722)	(0.196)
1:1	0.388	0.568	0.761	0.770	0.774	0.652	0.171
	(0.381)	(0.563)	(0.744)	(0.767)	(0.767)	(0.644)	(0.170)
1:2	0.254	0.432	0.730	0.736	0.721	0.576	0.219
	(0.240)	(0.440)	(0.731)	(0.736)	(0.720)	(0.572)	(0.223)
1:3	0.148	0.416	0.583	0.584	0.666	0.479	0.206
	(0.136)	(0.416)	(0.582)	(0.575)	(0.668)	(0.475)	(0.210)
Average	0.349	0.566	0.721	0.761	0.786		
	(0.343)	(0.576)	(0.716)	(0.758)	(0.784)		
S.D.	0.156	0.138	0.080	0.113	0.0986		
	(0.161)	(0.141)	(0.078)	(0.118)	(0.098)		

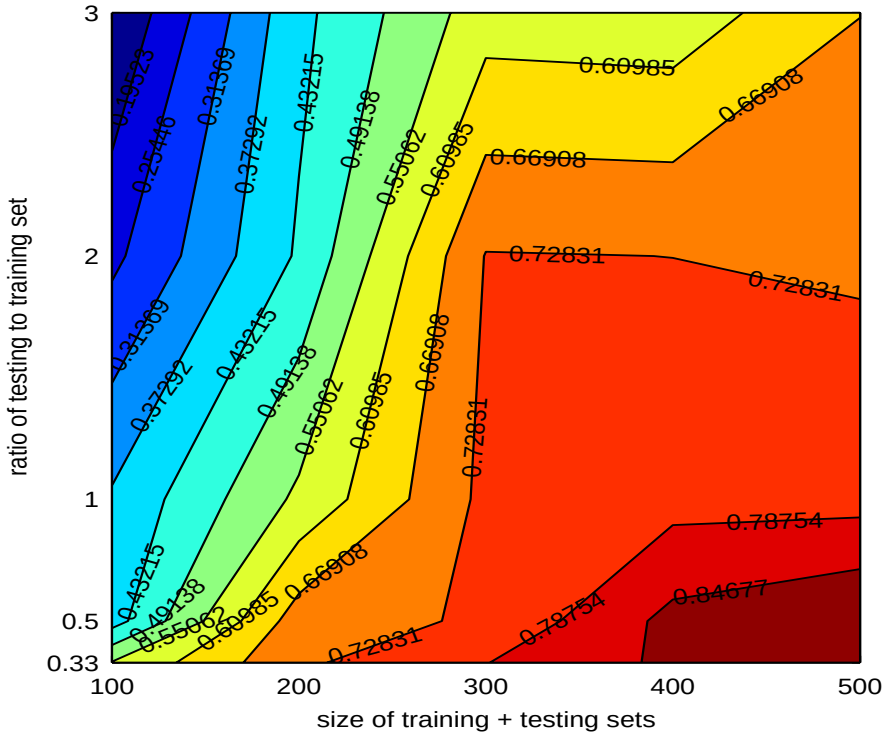

Figure 28: Contour plot of slopes with different sizes and ratios of training and testing set.

CHAPTER IV

MULTIPLE TESTING IN LARGE-SCALE CONTINGENCY TABLES

4.1. Introduction

One of the most common test procedures applied to two-way contingency tables is a test of independence (or association) between two categorizations. In general, the test of independence uses χ^2 tests or likelihood ratio tests that can be called "globally significant tests." The basic idea of these tests is as follows: If the sum of all the differences between observed and expected frequencies of all cells in a contingency table is small in a statistical sense, independence between two categorizations is accepted; if the sum of the differences is large, independence is rejected. However, the global tests do not accurately identify individual cells that significantly impact the final decision of rejecting the null hypothesis. The issue of identification of significant individual cells is especially important in the large-scale contingency tables where the number of cells $\gg 4$. Agresti (2002) pointed out several limitations of the global tests. He reviewed follow-up methods to global tests such as a partitioning of the χ^2 method as well as a method based on standardized and adjusted residual that allows further investigation of the associations in the contingency table. Partitioning of χ^2 is a method for exploring the associations by dividing the large tables into smaller ones. Lancaster (1949) showed that any r $\times$ c table can be reduced to (r-1) $\times$ (c-1) independent 2×2 tables. Hence, the interpretation of small tables is straightforward. Figure 29 illustrates how a 3×3 table is reduced to four 2×2 tables. In large-scale contingency tables, however, this method becomes too complicated as it generates

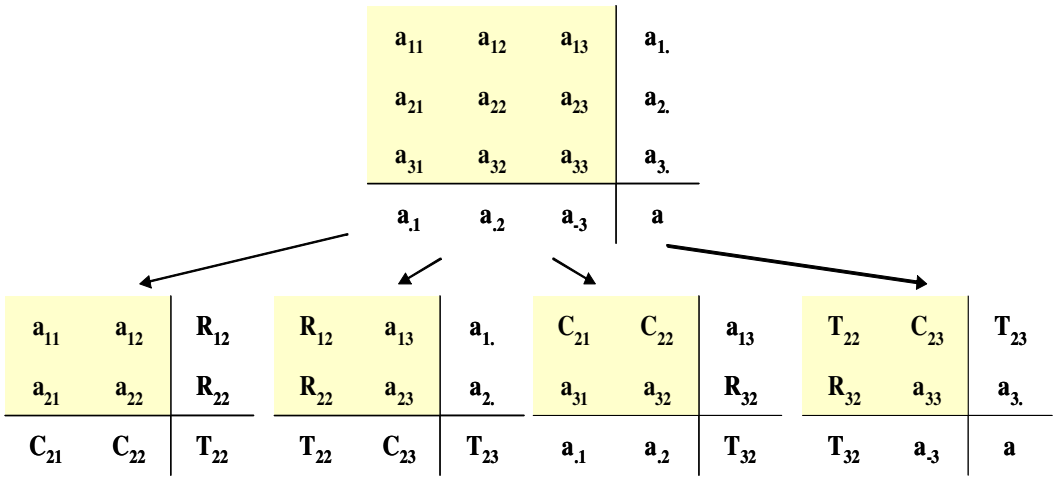

Figure 29: Partitioning of χ^2 of the 3×3 table.

too many 2×2 tables. For example, a 10×10 table produces 81 tables of 2×2 size, which makes the extraction of meaningful information cumbersome. Haberman (1973) defined the Standardized and Adjusted Residual (STAR) statistic for each cell and showed that this statistic is asymptotically standard normal under the null hypothesis of independence in each category. Therefore, the STAR statistics that are greater or less than a certain threshold indicate lack of fit to the null distribution in that cell (Agresti, 2002). The STAR method is simple but does not provide an objective way to determine a threshold since the threshold depends upon the number of degrees of freedom in the contingency table. Also, under the simultaneous consideration of all cell in the contingency table, the STAR method produces many false positives (Agresti, 2002). Another method was also introduced by Haberman (1973), who utilized a normal probability plot of STAR values that provides a nice graphical representation. However, the interpretation of a normal probability plot is frequently subjective, particulary when the number of cells to be tested is large. Therefore, there is a need for a method able to systematically and objectively identify the independence of each cell in the contingency tables. In this Chapter, we consider the problem of testing individual cells in a large-scale contingency table as a problem

of multiple testing.

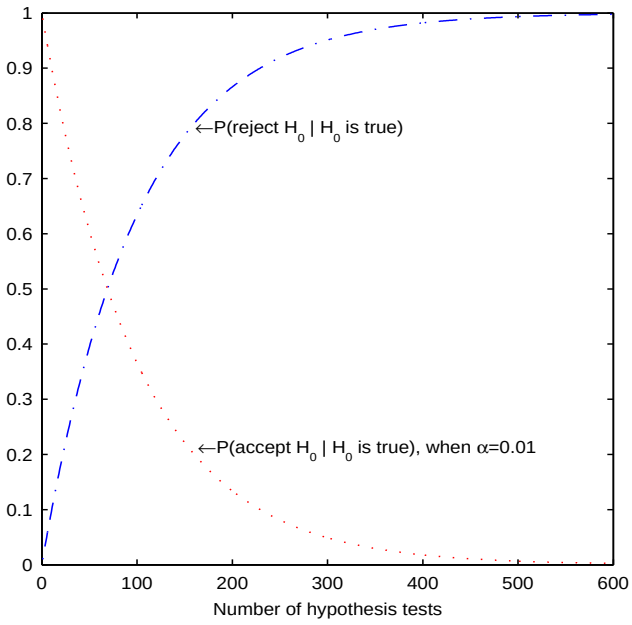

Figure 30: Illustration of multiplicity problems. False positive rates vs. the number of hypotheses. H_0 is null hypothesis.

In multiple testing problems, family-wise error rates have been used under simultaneous consideration to avoid the multiplicity effect. Figure 30 shows the multiplicity effect when α (i.e., P(reject $H_0|H_0$ is true)) is 0.01 for any single test. Applying the single testing procedure to the multiple testing problem leads to an exponential increase of false positive rates. More precisely, the probability that at least one of the tests leads to rejection of H_0 when H_0 holds, increases exponentially with the number of hypotheses. A convenient new definition of error rate, called false discovery rate (FDR) was proposed by Bejamini and Hochberg (1995). The FDR is the proportion of false positives among all the hypotheses rejected. The FDR has been used for microarray analysis to find co-expressed genes (Tusher *et al.* (2001), Efron *et al.* (2001), Efron and Tibshirani (2002), Dudoit *et al.* (2003)) as well as the genetic study to identify drugs causing mutations in the viral genome (Efron,

2004). Moreover, FDR has been applied to identify active voxels in neuroimaging data (Genovese *et al.* (2002), Wink *et al.* (2004)). In these studies, a hypothesis test is performed in each voxel whether the voxel contributes to classification between different experimental conditions. As an extension of original FDR, Storey (2002, 2003) and Storey *et al.* (2004) introduced the positive False Discovery Rate (pFDR) and Efron *et al.* (2001) proposed the Local False Discovery Rate (Local FDR). Moreover, the case when the hypotheses are dependent was considered by Yekutieli and Benjamini (1999) and Benjamini and Yekutieli (2001).

In this Chapter, we propose a procedure for testing independence of categories in individual cells of a contingency table based on a multiple testing framework. In addition, we perform simulation studies to compare the power of different multiple testing procedures in the contingency table. Finally, the proposed procedure is applied to identify the patterns of pair-wise associations of amino acids involved in β-sheet bridges.

4.2. Control Procedures in Multiple Testing

4.2.1 The Family-Wise Error Rate

In a multiple hypothesis test, assessing the number of false positives is necessary because a mere use of single inference procedures results in a significant number of false positives (Benjamini and Hochberg, 1995). Table 17 shows the possible outcomes from m hypothesis tests. The Family-Wise Error Rate (FWER), which has been classically used as a compound error rate in the setup of multiple hypothesis testing, is defined as the probability of generating one or more false rejections, i.e.,

$$\text{FWER} = \Pr\left[V \geq 1\right], \tag{17}$$

where V is the number of rejected hypotheses when the hypothesis is true. Shaffer (1995) summarized a variety of methods controlling the FWER. The most widely

Table 17: Outcomes from the multiple hypothesis tests of size m

	Accept null hypothesis	Reject null hypothesis	Total
True null hypothesis	U	V	m_0
False null hypothesis	T	S	m_1
Total	W	R	m

used one is the Bonferroni method. This method rejects H_i if $p_i \leq \alpha_i$, where p_i is the p-value of the ith hypothesis (i.e., H_i). In general, α_i is determined equally for all hypotheses (e.g., $\alpha_i = \frac{\alpha}{m}$). Therefore, the overall FWER is less than or equal to α. Other family-wise methods were developed to improve the power of the Bonferroni method, but they are still too stringent to detect false hypotheses. In other words, they can hardly reject the null hypothesis when it is actually false. In particular, the power significantly decreases as the number of hypotheses increases, where the power is the proportion of false null hypotheses which are correctly rejected.

4.2.2 The False Discovery Rate

Bejamini and Hochberg (1995) introduced the False Discovery Rate (FDR), defined as the expected proportion of false positives out of all rejected null hypotheses. The advantage of the FDR is to identify as many significant hypotheses as possible while keeping a relatively small number of false positives (Storey and Tibshirani, 2003). With a large family of hypotheses, the advantages over the FWER are substantial. In Table 17, R is the number of rejected null hypotheses, and V is the number of falsely rejected null hypotheses. Then the FDR is defined as

$$E\left[\frac{V}{V+S}\right] = E\left[\frac{V}{R}\right]. \tag{18}$$

Several important properties of the FDR were discussed in Benjamini and Hochberg (1995). For instance, R should be positive; if it is not, $\frac{V}{R}$ cannot be defined. A more

exact definition of the FDR is

$$E\left[\frac{V}{R}|R>0\right]P(R>0). \tag{19}$$

Understanding the relationship between the FDR and the FWER is important. When m_0=m, the FDR is equivalent to the FWER. When $m_0 < m$, FDR has more power in the sense that the FDR is less stringent in the multiple testing procedure. Benjamini and Hochberg (1995) proved that an ordered p-value method controls the specified FDR. This method is implemented as follows:

Consider a series of null hypotheses that are tested simultaneously,

$$H_1, H_2, \cdots, H_m.$$

We denote the corresponding independent test statistics, p-values, and ordered p-values as

$$Y_1, Y_2, \cdots, Y_m,$$

$$P_1, P_2, \cdots, P_m,$$

$$P_{(1)} \leq P_{(2)} \leq \cdots \leq P_{(m)}.$$

1. For a fixed α, where $0 \leq \alpha \leq 1$.
2. $\hat{i} = \max\left[i : P_{(i)} \leq \frac{i}{m} \cdot \frac{\alpha}{\pi_0}\right]$.
3. If $\hat{i} \geq 1$, $\Omega \in \{\text{All rejected } H_i \text{ with } P_i \leq P_{(\hat{i})}\}$ with FDR $(\Omega) \leq \alpha$.
   If $\hat{i} = 0$, Do not reject any hypothesis since $\Omega = \emptyset$.

Let $\pi_0(= \frac{m_0}{m})$ denote the proportion of true H_i. In general, $\pi_0 = 1$ is the most conservative possible choice. Several studies discussed the estimate of π_0 (Storey and Tibshirani, 2003; Efron, 2004). An example of Benjamini and Hochberg procedure is presented in Appendix C.

4.2.3 The Positive False Discovery Rate

Storey (2002, 2003) introduced the positive False Discovery Rate (pFDR). The term "*positive*" is included because it assumes that at least one significant hypothesis would occur.

$$\text{pFDR} = \text{E}\left[\frac{V}{R}|R > 0\right]. \tag{20}$$

In terms of the controlling procedure, the Storey procedure is different from the procedure of the Benjamini-Hochberg. The latter fixes α first and then derives a rejection rule (or decision rule) that achieves FDR $\leq \alpha$ while the former fixes the rejection rule first and then estimates FDR based on this rejection rule. A detailed description of Storey's procedure including estimation of the pFDR can be found in Storey (2002) or Appendix B. Here we describe the Storey's controlling procedure briefly.

1. Reject all H_i with $P_i \leq P_{(\hat{i})}$ such that $\hat{i} = \max\left[i : \hat{\text{pFDR}}_\lambda(P_{(i)}) \leq \alpha\right]$.

2. Estimate $\hat{\text{pFDR}}_\lambda(P_{(i)})$, which is less than α.

3. Thus,

$$\begin{aligned}
\hat{i} &= \max\left[i : \hat{\text{pFDR}}_\lambda(P_{(i)}) \leq \alpha\right] \\
&= \max\left[i : \frac{\hat{\pi_0}(\lambda) m P_{(i)}}{i} \leq \alpha\right] \\
&= \max\left[i : P_{(i)} \leq \frac{i}{m} \cdot \frac{\alpha}{\hat{\pi_0}(\lambda)}\right].
\end{aligned}$$

Here λ, a part of the estimate of π_0 (or $\hat{\pi}_0$), is determined via a tradeoff between bias and variance (Storey, 2003). Note that the Storey's procedure is the same as that of the Benjamini-Hochberg', except for estimating π_0. The relationship of the two procedures is described in Storey (2002) who has shown that the two procedures are equivalent when $\hat{\pi_0} = 1$. However, if $\hat{\pi_0} < 1$ and $\hat{\pi_0}$ can be properly estimated,

the Storey's procedure provides more power while controlling the same FDR. In other words, if the Storey's and Benjamini-Hochberg's procedures reject the same number of hypotheses, the Storey's procedure has a smaller FDR (Storey, 2002).

4.2.4 The Local False Discovery Rate

Efron *et al.* (2001) introduced the Local False Discovery Rate (Local FDR), the empirical Bayes version of the original FDR. Suppose the test statistics from multiple hypotheses follow a mixture distribution of two classes, i.e., statistics for true null and false null. Prior probabilities and their corresponding densities are represented as follows:

$$\begin{aligned} \pi_0 &= \text{probability of true null}, \quad f_0(y) = \text{the density of } Y \text{ for true null.} \\ \pi_1 &= \text{probability of false null}, \quad f_1(y) = \text{the density of } Y \text{ for false null.} \end{aligned}$$

Then the mixture density can be expressed as

$$f(y) = \pi_0 f_0(y) + \pi_1 f_1(y). \tag{21}$$

Given y, the posterior probabilities of being in either the true null class or the false null class are as follows:

$$\Pr\{\text{true null}|y\} = \frac{\pi_0 f_0(y)}{f(y)}, \tag{22}$$

$$\Pr\{\text{false null}|y\} = 1 - \Pr\{\text{true null}|y\} = \frac{\pi_1 f_1(y)}{f(y)}. \tag{23}$$

The Local FDR is defined to be

$$\text{Local FDR}(y) = \frac{\pi_0 f_0(y)}{f(y)}, \tag{24}$$

where $\pi_0 = 1$ gives the upper bound of the Local FDR. Efron (2004) suggested the following procedure to identify significant hypotheses.

1. Estimate $f(y)$ from test statistics, say $\hat{f(y)}$.

Table 18: A two-way $r \times c$ contingency table

	1	2	$\cdots$	c	
1	N_{11}	N_{12}	$\cdots$	N_{1c}	N_{1*}
2	N_{21}	N_{22}	$\cdots$	N_{2c}	N_{2*}
$\cdots$			$\cdots$		$\cdots$
r	N_{r1}	N_{r2}	$\cdots$	N_{rc}	N_{r*}
	N_{*1}	N_{*2}	$\cdots$	N_{*c}	N_{**}

2. Estimate a null density $f_0(y)$, say $\hat{f}_0(y)$.

3. Estimate π_0, say $\hat{\pi_0}$.

4. Compute the Local FDR$(y) = \frac{\hat{\pi_0}\hat{f_0(y)}}{\hat{f(y)}}$.

5. Declare y significant if Local FDR$(y) \leq \delta$, where δ is some threshold value.

The Local FDR, as its name suggests, provides a measure for the *specific (or local)* hypothesis by taking the ratio of true null density to mixture density for each set of test statistics. Thus, the small value of the ratio (that is, $f(y)$ is much larger than $\pi_0 f_0(y)$) implies a high chance that the hypothesis with statistic y is false. Efron and Tibshirani (2002) showed that the Local FDR has a close relationship with Benjamini and Hochberg's FDR. The conditional expectation of the Local FDR given a rejection region is the same as the Benjamini and Hochberg's FDR.

4.3. Multiple Testing in Contingency Tables

Our main interest is statistical inference of independence of categories in each cell in the contingency table. In this Chapter, we mainly discuss two-way contingency tables, but an extension to three-way or higher order tables can be made as well. Table 18 presents a two-way contingency table.

Usually, χ^2 tests or likelihood ratio tests have been used to identify the association of two categorizations under the null hypothesis of independence, i.e.,

$$H_0 : p_{ij} = p_{i*} \cdot p_{*j}, \tag{25}$$

$$i = 1, 2, \cdots, r, \quad j = 1, 2, \cdots, c.$$

$$X^2 = \sum_i \sum_j \frac{(N_{ij} - E_{ij})^2}{E_{ij}}, \; L^2 = 2 \sum_i \sum_j N_{ij} \log\left(\frac{N_{ij}}{E_{ij}}\right). \tag{26}$$

Pearson's χ^2 and the likelihood ratio test statistics, i.e., L^2, are defined in Equation 26. Here N_{ij} and E_{ij} are the observed and expected values in a cell corresponding to the ith row and the jth column. We call the χ^2 and the likelihood ratio tests *"globally significant tests"*, as these test statistics are derived from the sum of the deviations in all cells, $\sum_i \sum_j (N_{ij} - E_{ij})$. These global tests can evaluate overall association for the two categorizations in the contingency table but give little information about individual cells. Cochran (1954) and Berkson (1938) warned that the unguarded use of globally significant tests can mislead decision makers. For instance, if the χ^2 tests accept the null hypothesis, one should conclude that no significant association exists between two categorizations. However, some cells can have large deviations between N_{ij} and E_{ij} and the χ^2 tests fail to identify those cells containing useful information. Therefore, to find each important cell in the contingency table, appropriate methods need to be developed.

We consider now the contingency table as a data set to test multiple hypotheses simultaneously. More precisely, for an $r \times c$ contingency table, we have following $r \times c$ hypotheses.

$$\begin{aligned} H_1 : p_{11} &= p_{1*} \cdot p_{*1} \\ H_2 : p_{12} &= p_{1*} \cdot p_{*2} \\ &\cdots \\ H_{r\times c} : p_{rc} &= p_{r*} \cdot p_{*c}. \end{aligned}$$

Under the null hypothesis of independence, the adjusted residual (e_{ij}) for each cell can be defined as follows.

$$e_{ij} = \frac{N_{ij} - \frac{N_{i*} \cdot N_{j*}}{N_{**}}}{\left(\frac{N_{i*} \cdot N_{j*}}{N_{**}}\right)^{\frac{1}{2}}} = \frac{N_{ij} - E_{ij}}{(E_{ij})^{\frac{1}{2}}}. \tag{27}$$

Haberman (1973) proved that

$$e_{ij} \xrightarrow{\mathcal{D}} \mathcal{N}(0, v_{ij}), \tag{28}$$

where

$$v_{ij} = \left(1 - \frac{N_{i*}}{N_{**}}\right)\left(1 - \frac{N_{*j}}{N_{**}}\right). \tag{29}$$

However, asymptotic variance of e_{ij} is less than or equal to 1 unless the sample size is large enough (Haberman, 1973). A Standardized and Adjusted Residual (STAR), derived from dividing e_{ij} by its standard error, has been utilized as a corrected statistic. Under H_0, STAR values follow an asymptotic standard normal distribution.

$$\tilde{e}_{ij} = \frac{N_{ij} - E_{ij}}{(E_{ij} v_{ij})^{\frac{1}{2}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1). \tag{30}$$

The complete derivation can be found in Haberman (1973) and Agresti (2002). We used $\tilde{e}_{ij}$ as a test statistics for each cell in a contingency table. Agresti (2002) mentioned that the absolute value of $\tilde{e}_{ij}$, which exceeds about 2 (or in some cases 3), indicates the significant difference between observed and expected frequencies in that cell. Additionally, Haberman (1973) suggested the normal probability plotting of $\tilde{e}_{ij}$ values for identifying lack of independence of cells. Thus, the $\tilde{e}_{ij}$ that significantly deviates from the straight line is interpreted as an indicator of strong association between categories i and j. However, as we mentioned earlier, the methods described above are rather subjective thus do not provide an objective measure of large deviation. Below we propose a multiple testing procedures for the contingency tables to identify the individual cells that are significantly associated between categories. The

proposed procedure is summarized as follows:

Summary of the proposed procedure (multiple testing in the contingency table)

Consider a two-way $r \times c$ contingency table,

1. Construct $r \times c$ hypotheses for testing the independence of each cell.

$$H_{ij} : p_{ij} = p_{i*} \cdot p_{*j} \quad i = 1, 2, \cdots, r; \; j = 1, 2, \cdots, c.$$

2. Compute the corresponding test statistics based on STAR.

$$\tilde{e}_{ij} = \frac{N_{ij} - E_{ij}}{(E_{ij} v_{ij})^{\frac{1}{2}}} \quad i = 1, 2, \cdots, r; \; j = 1, 2, \cdots, c.$$

   Under the null hypothesis, $\tilde{e}_{ij} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1)$.

3. Choose one of the multiple testing procedures described in Section 2 to identify the significant cells in a contingency table.

4.4. Simulation Studies

4.4.1 The Setting

In this section we apply our proposed procedure to the simulated data set. We compare the empirical power, type I error, and false discovery rate of four different multiple testing procedures and the individual test with the corresponding false positive rate. Details are as follows:

1. Bonferroni procedure controlling FWER at 0.01.

2. Bejamini and Hochberg procedure controlling FDR at 0.01.

3. Storey procedure controlling at pFDR at 0.01.

4. Efron procedure with the threshold of local FDR at 0.01.

5. Individual test with the p-value threshold at 0.01.

To clarify, we define the empirical power, type I error, and false positive rate as follows:

$$\text{Empirical power} = \frac{\text{\# of correcly rejected hypotheses}}{\text{\# of false null hypotheses}},$$

$$\text{Empirical type I error} = \frac{\text{\# of incorrectly rejected hypotheses}}{\text{\# of true null hypotheses}},$$

$$\text{Empirical FDR} = \frac{\text{\# of incorrectly rejected hypotheses}}{\text{\# of rejected hypotheses}}.$$

In this study, we consider a 4×4 contingency table. That is, 16 hypotheses (=total number of cells) are considered for testing independence of categories in each cell. Thus, the family of null hypotheses states that 4×4 categories in the contingency table are independent.

$$H_{ij} : p_{ij} = p_{i*} \cdot p_{*j} \quad i = 1, 2, 3, 4, \quad j = 1, 2, 3, 4.$$

Each individual null hypothesis is tested based on standardized and adjusted residual statistics (Equation 30), and these test statistics are assumed to be independent.

The proportion of true null hypotheses out of 16 is set to be 0%, 25%, 50%, and 75%. In other words, the number of significant ones out of a total of 16 hypotheses is 16, 12, 8, and 4, respectively. We shall describe the way to determine the proportion of true null hypotheses in the next paragraph. In addition, we define θ (Equation 31), which measures a magnitude of difference between p_{ij} and $p_{i*} \cdot p_{*j}$ in the contingency table. Hence, the contingency table having large θ implies the huge

Table 19: A 4×4 contingency table containing joint probabilities (p_{ij}) and marginal probabilities (p_{i*} and p_{*j})

0.115	0.201	0.044	0.012	0.372
0.034	0.142	0.029	0.159	0.364
0.025	0.091	0.024	0.017	0.157
0.008	0.049	0.024	0.026	0.107
0.182	0.483	0.121	0.214	1

Table 20: A 4×4 contingency table containing the probabilities computed by the product of marginal probabilities (i.e. $p_{i*} \cdot p_{*j}$) in Table 19

0.068	0.180	0.045	0.080
0.066	0.176	0.044	0.078
0.029	0.076	0.019	0.034
0.019	0.052	0.013	0.023

discrepancy between observed and expected frequencies in that contingency table. In our simulations, different θs are considered in each scenario, where

$$\theta = \sum_{i=1}^{r} \sum_{j=1}^{c} |p_{ij} - p_{i*} \cdot p_{*j}| \quad . \tag{31}$$

Finally, we consider sample sizes of $n = 100, 500$, and 1000 to investigate their effects.

Let's explain more precisely how we determine the proportion of true null hypothesis in each scenario of the simulation. The first step is to obtain (or generate) the proportion of each cell in the 4×4 contingency table (Table 19). Here, we obtain the contingency table from the example, described on page 80 of Agresti (2002). In this table, the probabilities in the right most column and the bottom row indicate the marginal probabilities. The next step is to construct the table (Table 20) containing the probabilities computed by the product of marginal probabilities of the contingency table in Table 19. For example, in the cell at the intersection of the first row and the first column, $0.068 = 0.372 \times 0.182$, in the cell at the intersection of the first row and the second column, $0.180 = 0.372 \times 0.483$, and so on. Note that the

probabilities of all cells between two tables are different. This implies that all cells in the contingency table in Table 19 violate the null hypothesis of independence (0% null). We call this contingency table an initial table. Put another way, the initial table should not contain the cells that conform to the null hypothesis.

$$\text{Initial table: } H_{ij} : p_{ij} \neq p_{i*} \cdot p_{*j} \quad i = 1, 2, 3, 4, \quad j = 1, 2, 3, 4.$$

From the initial table, each iteration is performed to force some specific cells to agree with the null hypothesis. Table 21 shows four iterations from the initial table where each iteration increases the number of cells conforming to the null hypothesis. The values inside a parenthesis represent the cells, which satisfy the null hypothesis ($p_{ij} = p_{i*} \cdot p_{*j}$). For instance, after the first iteration, the table contains four cells conforming to the null hypothesis (25% true null) and after the second iteration, the table contains eight cells conforming to the null hypothesis (50% true null). Finally, after the fourth iteration, the table contains twelve cells, which conform to the null hypothesis (75% true null). Also, in order to change the θ in each iteration, we add or subtract the numbers (probabilities) in the cells conforming to the null hypothesis while keeping the same marginal sums of the contingency table. We then utilize the probabilities in the contingency table obtained from each iteration for sample size ($= n$) to generate multinomial random numbers, which allow the multiple testing procedures to calculate the power, type I error, and the FDR.

4.4.2 Results

Each simulation is done with 5000 repetitions. Figures 39 $\sim$ 41 in Appendix D present all the simulation results. Each table is obtained under the different simulation setup as previously illustrated.

Figures 31 $\sim$ 34 illustrate the average empirical power, type I error, and FDR for the five different procedures (i.e., individual, Bonferroni, Benjamini-Hochberg,

Table 21: Iterations to specify the proportion of true null in the contingency table

Initial table

0.115	0.201	0.044	0.012
0.034	0.142	0.029	0.159
0.025	0.091	0.024	0.017
0.008	0.049	0.024	0.026

Iteration 1

0.098	0.150	0.025	0.099
0.036	0.205	0.064	0.059
(0.029)	(0.076)	0.020	0.032
(0.019)	(0.052)	0.012	0.024

Iteration 2

(0.068)	(0.180)	0.005	0.119
(0.066)	(0.176)	0.084	0.038
0.009	0.095	(0.019)	(0.034)
0.040	0.031	(0.013)	(0.023)

Iteration 3

(0.068)	(0.180)	(0.045)	(0.080)
(0.066)	(0.176)	(0.044)	(0.078)
(0.029)	(0.076)	0.002	0.050
(0.019)	(0.052)	0.030	0.006

Storey, and Efron). In each panel of Figures, the x-axis is a different value of θ and the y-axes are respectively the average empirical power, type I error, and FDR. The followings observations are made.

1. Figure 31 shows that the power of all procedures increases when both sample size and θ increase.

2. Figure 31 shows that in general, the individual test produces larger power as well as higher type I error and false discovery rate than the other four procedures. However, as the proportion of true null hypotheses decreases (i.e., the proportion of significant hypotheses increases), the FDR-related procedures give the power comparable to the individual test. Storey's procedure yields larger power than the individual test in some cases. We explain this more clearly in Section 4.4.3. For the comparison of multiple testing procedures, the power is uniformly ranked as follows:

   Storey > Benjamini-Hochberg > Efron > Bonferroni.

3. Figure 33 shows that the FDR decreases when the proportion of true null decreases. The reason is described as follows: Using the notation of Table 17, let the number of true and false null be denoted by m_0 and m_1. Moreover, let type I error and power be denoted by α and $(1-\beta)$. Then the FDR can be represented as follows:

$$FDR = \frac{m_0\alpha}{m_0\alpha + m_1(1-\beta)}. \tag{32}$$

   Thus, if α and $(1-\beta)$ are fixed, the FDR decreases as m_0 decreases. Note that as m_0 decreases, m_1 increases.

4. Figure 32 shows that type I error increases as the number of true null decreases

when the FDR is fixed. From Equation 32, we can derive the following equation,

$$\alpha = \frac{m_1}{m_0} \cdot \frac{FDR}{1 - FDR} \cdot (1 - \beta). \tag{33}$$

Equation 33 shows that α is inversely proportional to m_0.

5. Another interesting observation on type I error indicates that as the proportion of true null decreases, the magnitude of difference between the procedures becomes small.

6. Figure 34 shows type I error and false discovery rate when the proportion of true null hypothesis is 100%. In this case, the power is not defined by its definition. The individual test produces larger type I error and the FDR than other procedures over the different sample sizes.

4.4.3 Simulation with Normal Random Variable

In order to understand the behavior of different multiple testing procedures more clearly, we perform the simulation studies assuming a normal random variable. We perform 500 hypothesis tests of $\mu = 0$ *vs.* $\mu = \delta$ ($\delta = 1, 2, 3, 4$) for independent random variables $z_i \sim \mathcal{N}(\mu, 1), i = 1, \cdots, 500$, over 500 iterations. The null hypothesis for each test is $\mu = 0$. The proportion of null hypotheses (i.e., π_0) is set differently (i.e., $\pi_0 = 0.002\%, 0.2\%, 0.4\%, 0.6\%, 0.8\%, 0.9\%$). For each test the p-value is computed as $p_i = 2 \times P\{X \geq |z_i|\}$, where $X \sim \mathcal{N}(0, 1)$. The rejection region is chosen $\alpha = 0.01$. Figures 42 and 43 in Appendix E show the results of the simulations. Figures 35 and 36 also illustrate the results graphically. The overall results indicate that the power increases when δ increases. The power of the individual test is larger than that of other procedures. However, when the proportion of true hypotheses decreases (i.e., $\pi_0 \to 0$) and δ is large, the Storey's procedure controlling the pFDR renders larger power than the individual test procedure. The reason is as follows: For a p-value

threshold of t, the estimated pFDR is

$$\frac{m\hat{\pi}_0 t}{\#[p\text{-value} \leq t]},$$

where m is the number of hypothesis tests and $\hat{\pi}_0$ is the proportion of true null hypotheses. If $\frac{m\hat{\pi}_0}{\#[p\text{-value} \leq t]} < 1$, then the FDR threshold is more liberal than the individual test threshold, t (Storey, 2002). As t gets close to 1, we have $\#[p\text{-value} \leq t]$ about equal to m, and $\hat{\pi}_0 < 1$. Therefore, if $\hat{\pi}_0 < 1$, then some FDR threshold leads to more significant hypotheses (rejected null hypotheses) than the corresponding p-value threshold. Moreover, we observe that when π_0 is close to 1, the Benjamini and Hochberg's procedure is the same as the Storey's procedure.

Type I error of the individual test procedure and the Bonferroni procedure are constant over each simulation scenario. Type I error of the FDR-related procedures increases as the proportion of significant hypotheses increases (i.e., $\pi_0 \to 0$). Also as the δ increases, type I error of the FDR-related procedures increases.

With regards to the FDR, the individual test produces larger FDR than the other procedures when δ is small and the proportion of significant hypotheses increases.

We conclude that when the number of significant hypotheses is small, the individual test procedure gives large power but renders high FDR. However, if the number of significant hypotheses is large, the Storey's procedure produces large power with relatively small type I error compared to the individual test procedure.

4.5. Inferring Pair-Wise Amino Acid Patterns in β-Sheets

This section demonstrates the effectiveness of multiple testing in a contingency table through the identification of patterns of pair-wise association of amino acids involved in β-sheet bridges. When predicting the secondary protein structure, the prediction

rate of β-sheets is significantly lower than that of α-helices and loops due to algorithmic inability to capture the pair-wise associations. Thus, investigating the pair-wise associations in β-sheets can improve overall prediction accuracy of protein secondary structure as well as provide useful information of prediction of protein tertiary structure. In the last several decades many studies have addressed this issue. Von Heijne and Blomberg (1977, 1978) studied the pair correlations in β-strands among hydrophobic, neutral, and polar classes of residues. They revealed that residues within the same classes occur more often than expected by random chance. Lifson and Sander (1980) analyzed the frequencies of amino acid pairs in parallel and antiparallel structures and uncovered the number of trends in favored amino acid pairs. But their studies were performed on the group level so the results did not provide individual patterns of pair-wise association. Recent studies focused on antiparallel β-sheets (Wouters and Curmi (1995), Smith and Regan (1995), Hutchison *et al.* (1998)). They investigated two distinct sites based on the existence of hydrogen bonding in backbone NH and C=O: hydrogen bonded and non-hydrogen bonded. Their work revealed that two sites have different patterns of residue pairs. Zhu and Braun (1999) analyzed the propensity of residue pairs shifted by one of two residues along the strands from the nearest contacts and found preferential occurrence of tri-peptides and their favorite partners. In their study, the statistical contact energy between two amino acids was defined as the measure of propensity. General consensus of previous studies implies that nonrandom patterns of pair-wise associations exist across neighboring β-strands. This section contains the following contents:

- Section 4.5.1 illustrates the database.
- Section 4.5.2 describes the statistical formulation of the problem.
- Section 4.5.3 gives the results of pattern recognition of grouped amino acid pairs involved in β-sheet bridges.

- Section 4.5.4 provides the results of pattern recognition of individual amino acid pairs involved in β-sheet bridges.

4.5.1 Database

A set of 613 nonhomologous proteins listed in the December 2003 Protein Data Bank (PDB; http://www.rcsb.org/pdb/) were used. No pair among all the selected proteins had more than 33% identical residues. Secondary structures were assigned to the 613 proteins using the Definition of Secondary Structure Assignment algorithm (DSSP; http://www.cmbi.kun.nl/gv/dssp/) designed by Kabsch and Sander (1983). The sample format of the database is shown in Appendix E. The frequencies of the pairs in both parallel and antiparallel strands were obtained when two residues involved a bridge (Kabsch and Sander, 1983). Bridges were defined as follows: Suppose there are two nonoverlapping strings of the three residues such as i-1, i , i+1 and j-1, j, j+1.

A parallel bridge (i, j) = [Hydrogen-bond (i-1, j) **and** Hydrogen-bond (j, i+1)] **or** [Hydrogen-bond (j-1, i) **and** Hydrogen-bond (i, j+1)].

An antiparallel bridge (i, j) = [Hydrogen-bond (i, j) **and** Hydrogen-bond (j, i)] **or** [Hydrogen-bond (i-1, j+1) **and** Hydrogen-bond (j-1, i+1)].

Table 22 shows a summary of the database. Note that pair-wise associations in antiparallel strands are more common than they are in parallel strands, which implies that many β-sheets are formed in antiparallel fashion.

Table 22: Summary of data set

	Num. of proteins	Num. of pairs	Num. of pairs per protein
Parallel	613	14,064	23
Antiparallel	613	37,400	61

4.5.2 Statistical Formulation

The frequency of pair-wise associated amino acids can be modelled by a multinomial distribution. Suppose the experiment consists of $\mathcal{T}$ independent trials. The results of each trial is one of $\mathcal{C}$ mutually exclusive categories. Let random variable N_i count the number of occurrences of the ith outcome and p_i be the corresponding probability. Then the random vector $\mathbf{N} = (N_1, N_2, \cdots, N_{\mathcal{C}})$ has a multinomial distribution with parameters $\mathcal{T}$ and $\mathbf{p} = (p_1, p_2, \cdots, p_{\mathcal{C}})$. Then

$$f(N_1 = n_1, N_2 = n_2, \cdots, N_{\mathcal{C}} = n_{\mathcal{C}}) = \mathcal{T}! \prod_{i=1}^{\mathcal{C}} \frac{p_i^{n_i}}{n_i!}. \tag{34}$$

Each n_i is a nonnegative integer satisfying $\sum_{i=1}^{\mathcal{C}} n_i = \mathcal{T}$.

There are 20 different amino acids and thus the possible number of pairs are 400 ($= 20 \times 20$). However, we only consider 210 ($= \frac{20^2+20}{2}$) pairs since we do not distinguish between the two different types of amino acids in a pair. Let $N_{\cdot\cdot}$ denote the total number of pair-wise amino acids in the database. Then $\mathbf{N} = \underbrace{(N_{AA}, N_{AC}, \cdots, N_{YY})}_{210}$ has a multinomial distribution with parameters $N_{\cdot\cdot}$ and $\mathbf{p} = (p_{AA}, p_{AC}, \cdots, p_{YY})$. The observed frequency of associated pairs with amino acids type X and Y are denoted as n_{XY}. Note that n_{XY} and n_{YX} are counted in the same category. In addition, we denote n_{X*} or n_{*X} as the number of pair-wise associations with X and any other

amino acids. Then we have the following simple relationships:

$$\sum_{Y} n_{XY} = n_{X*}, \quad \sum_{X} n_{X*} = n_{**}.$$

Table 23 is the representation of the model via a two-way symmetric contingency table. Our main task is to identify which of the 210 pairs are significantly associated

Table 23: A two-way 20×20 contingency table containing the frequency of pair-wise amino acid in β-sheet bridges

	A	C	$\cdots$	Y	
A	N_{AA}	N_{AC}	$\cdots$	N_{AY}	N_{A*}
C	N_{CA}	N_{CC}	$\cdots$	N_{CY}	N_{C*}
$\cdots$			$\cdots$		$\cdots$
Y	N_{YA}	N_{YC}	$\cdots$	N_{YY}	N_{Y*}
	N_{*A}	N_{*C}	$\cdots$	N_{*Y}	N_{**}

to form β-sheet bridges. In each category of the contingency table, the null hypothesis is that two amino acids are paired at random. More precisely, we can construct the following 210 hypotheses.

$$\begin{aligned} H_1 : p_{AA} &= p_{A*} \cdot p_{*A}, \\ H_2 : p_{AC} &= p_{A*} \cdot p_{*C}, \\ &\cdots \\ H_{210} : p_{YY} &= p_{Y*} \cdot p_{*Y}. \end{aligned}$$

4.5.3 Pattern Recognition of Grouped Amino Acid Pairs

In this Section we investigate the associated patterns of pair-wise amino acids involved in β-sheet bridges at the group level. Twenty amino acids can be grouped based on their chemical properties. Chemical properties cause associations between amino acids that determine the structure of the folded protein and its biological function. The followings are the four groups of amino acids (Alberts *et al.* 1997):

- Negatively charged polar: Asp(D), Glu(E).
- Positively charged polar: Arg(R), Lys(K), His(H).
- Uncharged Polar: Asn(N), Gln(Q), Ser(S), Thr(T), Tyr(Y).
- Nonpolar: Ala(A), Gly(G), Val(V), Lue(L), Ile(I), Pro(P), Phe(F), Met(M), Trp(W), Cys(C).

Table 24: Associated patterns of grouped amino acid pairs in parallel strands (*: Estimated expected frequencies. +: Standardized and adjusted residuals)

	neg.cha.polar	pos.cha.polar	uncha.polar	nonpolar	Total
neg.cha.polar	38 $(40)^*$ $(-0.26)^+$	180 (58) (17.05)	167 (143) (2.34)	361 (506) (-11.64)	746
pos.cha.polar	180 (58) (17.05)	114 (86) (3.27)	289 (210) (6.30)	517 (745) (-15.35)	1,100
uncha.polar	167 (143) (2.34)	289 (210) (6.30)	822 (513) (16.84)	1,409 (1,821) (-18.91)	2,687
nonpolar	361 (506) (-11.64)	517 (745) (-15.35)	1,409 (1,821) (-18.91)	7,244 (6,459) (30.30)	9,531
Total	746	1,100	2,687	9,531	14,064

Tables 24 and 25 show the 4×4 contingency tables containing the observed and expected frequencies of grouped pairs and corresponding STAR values obtained from parallel and antiparallel strands. Most of the STAR values are large since the difference between the observed and expected frequencies tends to be inflated with large sample size. Figure 37 shows the scatter plot of STAR values between parallel and antiparallel strands. G1 $\sim$ G10 represent the indices of grouped amino acid pairs, defined in Table 26. This figure allows us to see the relative importance of the STAR values. Three clusters are observed. The grouped pairs in the upper and right (i.e.,

Table 25: Associated patterns of grouped amino acid pairs in antiparallel strands: (*: Estimated expected frequencies. +: Standardized and adjusted residuals)

	neg.cha.polar	pos.cha.polar	uncha.polar	nonpolar	Total
neg.cha.polar	220 (205)* (1.12)+	701 (325) (23.06)	816 (717) (4.46)	1,033 (1,523) (-19.43)	2,770
pos.cha.polar	701 (325) (23.06)	462 (515) (-2.65)	1,392 (1,136) (9.38)	1,834 (2,413) (-18.68)	4,389
uncha.polar	816 (717) (4.46)	1,392 (1,136) (9.38)	3,128 (2,507) (16.74)	4,347 (5,323) (-23.15)	9,683
nonpolar	1,033 (1,523) (-19.43)	1,834 (2,413) (-18.68)	4,347 (5,323) (-23.15)	13,344 (11,300) (42.69)	20,558
Total	2,770	4,389	9,683	20,558	37,400

Table 26: Grouping index

Association within or between groups	Index	Remark
neg.cha.polar : neg.cha.polar	G1	Within Group (WG)
neg.cha.polar : pos.cha.polar	G2	Between Group (BG)
neg.cha.polar : uncha.polar	G3	BG
neg.cha.polar : nonpolar	G4	BG
pos.cha.polar : pos.cha.polar	G5	WG
pos.cha.polar : uncha.polar	G6	BG
pos.cha.polar : nonpolar	G7	BG
uncha.polar : uncha.polar	G8	WG
uncha.polar : nonpolar	G9	BG
nonpolar : nonpolar	G10	WG

G10, G2, and G8) show strong association in both parallel and antiparallel strands. On the contrary, the grouped pairs in the lower and left (i.e., G7, G9, and G4) show strong disassociation in both strands. The grouped pairs in the middle part of Figure 37 (i.e., G6, G3, G1, and G5) do not exhibit significant association to form β-sheet bridges. Generally, the amino acids within a group (nonpolar vs. nonpolar

Table 27: Significance of grouped residue pairs in parallel strand: STAR: Standardized and adjusted residual. S: Significantly associated pair. N: Not significantly associated pair

Index	Grouped amino acid pair	STAR	p-value	BF	B-H	EF	ST
G10	nonpolar : nonpolar	30.3	0	S	S	S	S
G2	neg.cha.polar : pos.cha.polar	17.05	0	S	S	S	S
G8	uncha.polar : uncha.polar	16.84	0	S	S	S	S
G6	pos.cha.polar : uncha.polar	6.3	0	S	S	S	S
G9	uncha.polar : nonpolar	-18.91	0	S	S	S	S
G7	pos.cha.polar : nonpolar	-15.35	0	S	S	S	S
G4	neg.cha.polar : nonpolar	-11.64	0	S	S	S	S
G5	pos.polar : pos.polar	3.27	0.0012	N	S	S	S
G3	neg.cha.polar : uncha.polar	2.34	0.0193	N	N	N	S
G1	neg.cha.polar : neg.cha.polar	-0.26	0.7949	N	N	N	N

Table 28: Significance of grouped residue pairs in antiparallel strand. See the caption of Table 27 for definition of columns.

Index	Grouped amino acid pair	STAR	p-value	BF	B-H	EF	ST
G10	nonpolar : nonpolar	42.69	0	S	S	S	S
G2	neg.cha.polar : pos.cha.polar	23.06	0	S	S	S	S
G8	uncha.polar : uncha.polar	16.74	0	S	S	S	S
G6	pos.cha.polar : uncha.polar	9.38	0	S	S	S	S
G3	neg.cha.polar : uncha.polar	4.46	0	S	S	S	S
G9	uncha.polar : nonpolar	-23.15	0	S	S	S	S
G4	neg.cha.polar : nonpolar	-19.43	0	S	S	S	S
G7	pos.cha.polar : nonpolar	-18.68	0	S	S	S	S
G5	pos.cha.polar : pos.cha.polar	-2.65	0.00804	N	N	N	S
G1	neg.cha.polar : neg.cha.polar	1.12	0.26272	N	N	N	N

and uncharged polar vs. uncharged polar) are more likely to associate with themselves whereas the amino acids between groups (polar vs nonpolar) are not likely to associate with each other. Strong associations are observed between two differently charged groups (positively charged polar and negatively charged polar groups). It is interesting to observe that that in some cases, associated patterns are manifested differently between parallel and antiparallel strands (i.e., G1 and G5). For instance, amino acids in a positively charged polar group are likely to interact themselves in

parallel strands but not in antiparallel strands.

Tables 27 and 28 present the results of multiple testing procedures (Bonferroni (BF), Benjamini-Hochberg (B-H), Efron (EF), and Storey (ST)) for the grouped amino acids from parallel and antiparallel strands. The Storey's procedure identify the largest number of significantly associated grouped amino acids and the Bonferroni's procedure the least. However, since the large frequencies in each cell (due to grouping) lead to large STAR values in most of cases, it is dangerous to make a conclusion based solely on the multiple testing results. One remedial measure is shown in Figure 37 since it provides the relative significace. But this grouping structure is still limited as it cannot provide the specific patterns of individual pairs. The following Section studies the pattern of individual amino acids to understand the specific behavior of pair-wise association in β-sheet bridges.

4.5.4 Pattern Recognition of Individual Amino Acid Pair

Identifying patterns of individual amino acid pairs involved in β-sheet bridges is important for a better understanding of the rules of protein secondary structure formation. The primary aim of this application is to find "*favored pairs*" and "*unfavored pairs*" among the 210 possible pairs of amino acids. Here the terms "*favored*" or "*unfavored*" means that two amino acids like or do not like to be associated with each other to form β-sheet bridges. First, we present the multiple testing results and then interpret the results with the knowledge of chemical properties of amino acids.

• Multiple Testing Results

The results from the multiple testing procedures are reported in Tables 29 $\sim$ 32 and summarized in Table 33. Comparing the individual test procedure with the Bonferroni's procedure, the former with the p-value threshold of 0.01 per each hypothesis found more significant hypotheses than the latter controlling FWER=0.01. Among

the procedures controlling the FDR, the Storey's procedure found the largest number of hypotheses. The Benjamini-Hochberg and Efron procedures found the second and the third largest. Another observation is that the Storey's procedure identifies more significant hypotheses than the individual test procedure. This was explained in the simulations described in Section 4.4.3. In our example, the estimated proportions of true null ($\hat{\pi}_0 s$) of the parallel and antiparallel strands are 0.2433 and 0.1818, obtained from software developed by Storey (http://faculty.washington.edu/ jstorey/). In other words, the estimated proportion of significant hypotheses is large in our example. In this case, we prefer the Storey's procedure since the simulation study showed that the Storey's procedure produces large power with relatively small type I error when $\hat{\pi}_0$ is small.

Another important issue between the individual test procedure and the FDR-related procedures is an interpretation. For instance, in our β-sheet problem, we are more interested in the fraction of false positives among all rejected hypotheses than the probability of making one or more false positive rates.

To clarify, consider the case of significant pairs (both favored and unfavored) in antiparallel strands. The individual test found 118 significant pairs with the probability that at least one false positive is 0.88 . For the Bonferroni test, 76 pairs are found to be significant with the probability that at least one false positive is 0.01. The other three methods controlling the FDR can be interpreted similarly. For example, Storey's procedure declares 130 significant pairs among which the proportion of false positives is 0.01. Note that Storey's procedure found more significant hypotheses (large power) than do the procedures of Benjamini and Hochberg and the Efron while controlling the same FDR.

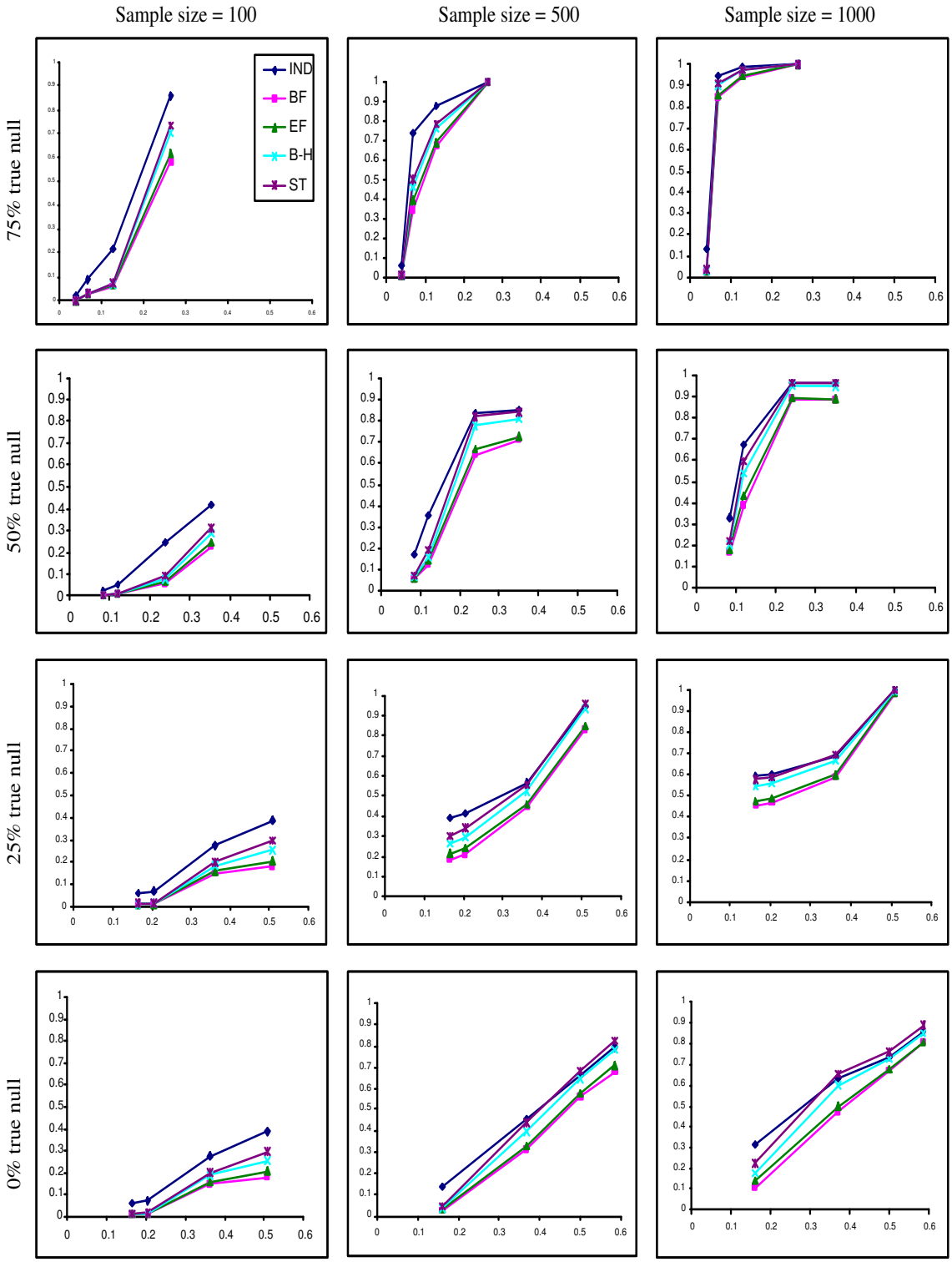

Figure 31: Average empirical power: the individual test ($\diamond$), the Bonferroni ($\square$), Efron ($\triangle$), the Benjamini and Hochberg($\times$), the Storey ($*$).

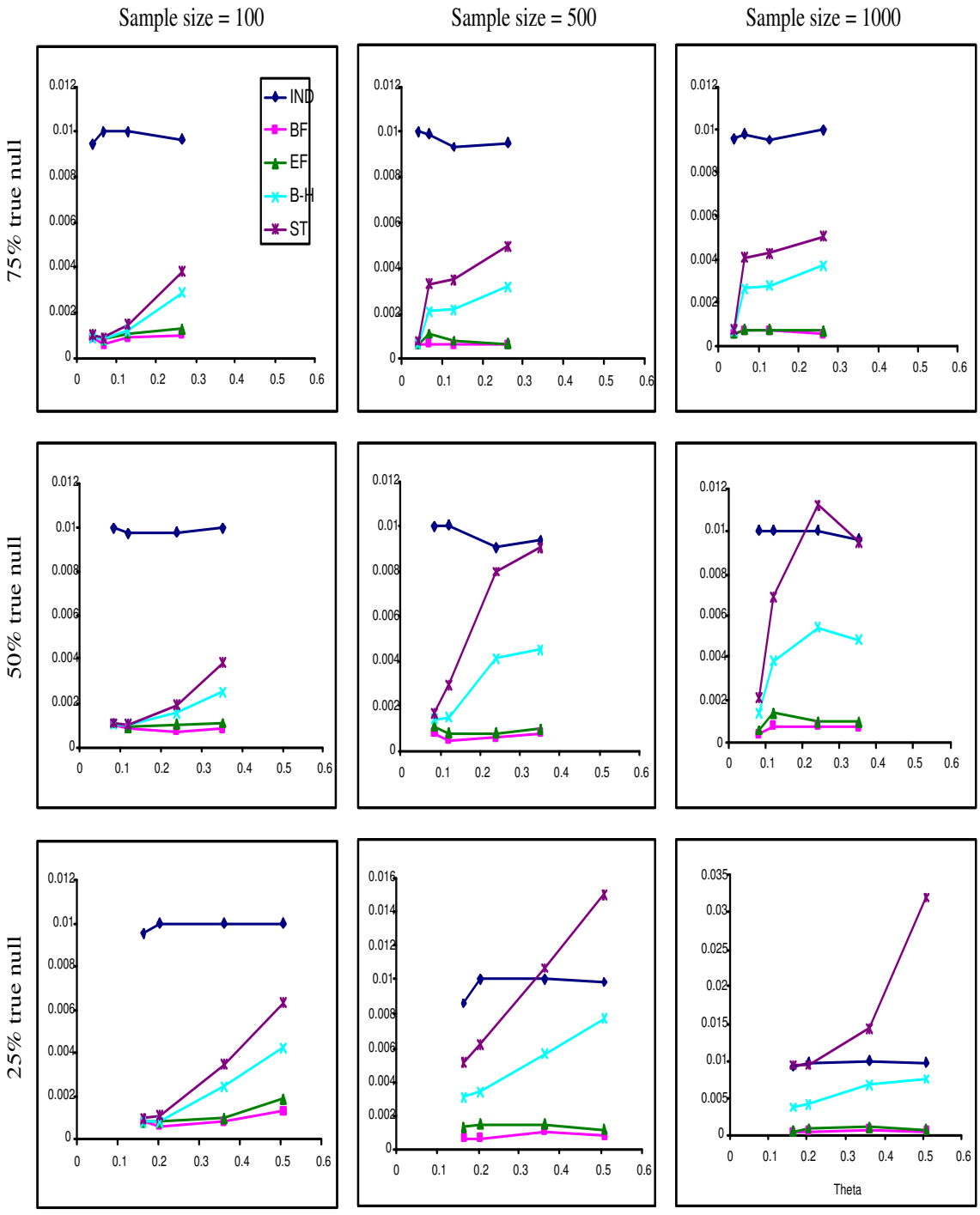

Figure 32: Average empirical type I error: See the caption of Figure 31 for the illustration. Note that type I error is not defined when the proportion of true null hypothesis is 0%.

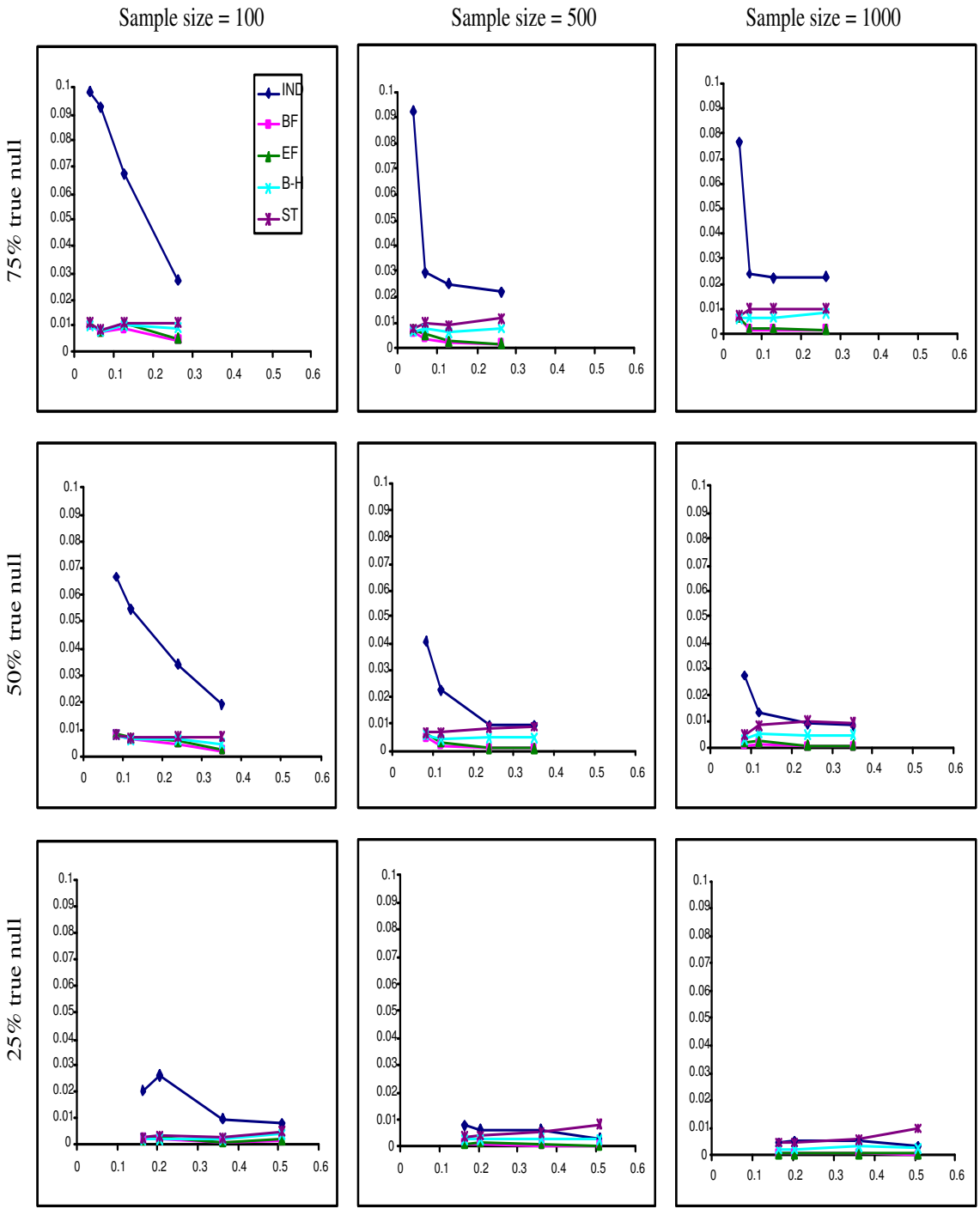

Figure 33: Average empirical false discovery rate. See the caption of Figure 31 for the illustration. Note that false discovery rate is 0 when the proportion of true null hypothesis is 0%.

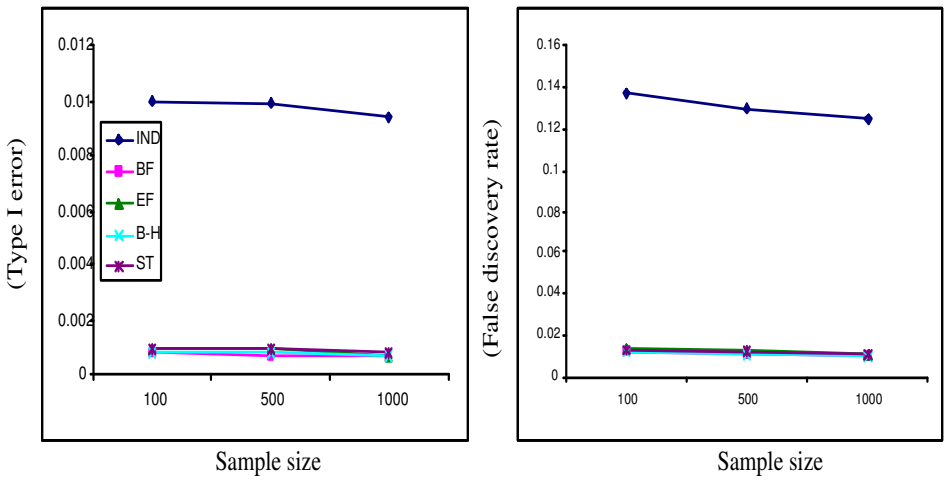

Figure 34: Average empirical type I error and false discovery rate when the proportion of true null hypothesis is 100%. Note that power is not defined in this case.

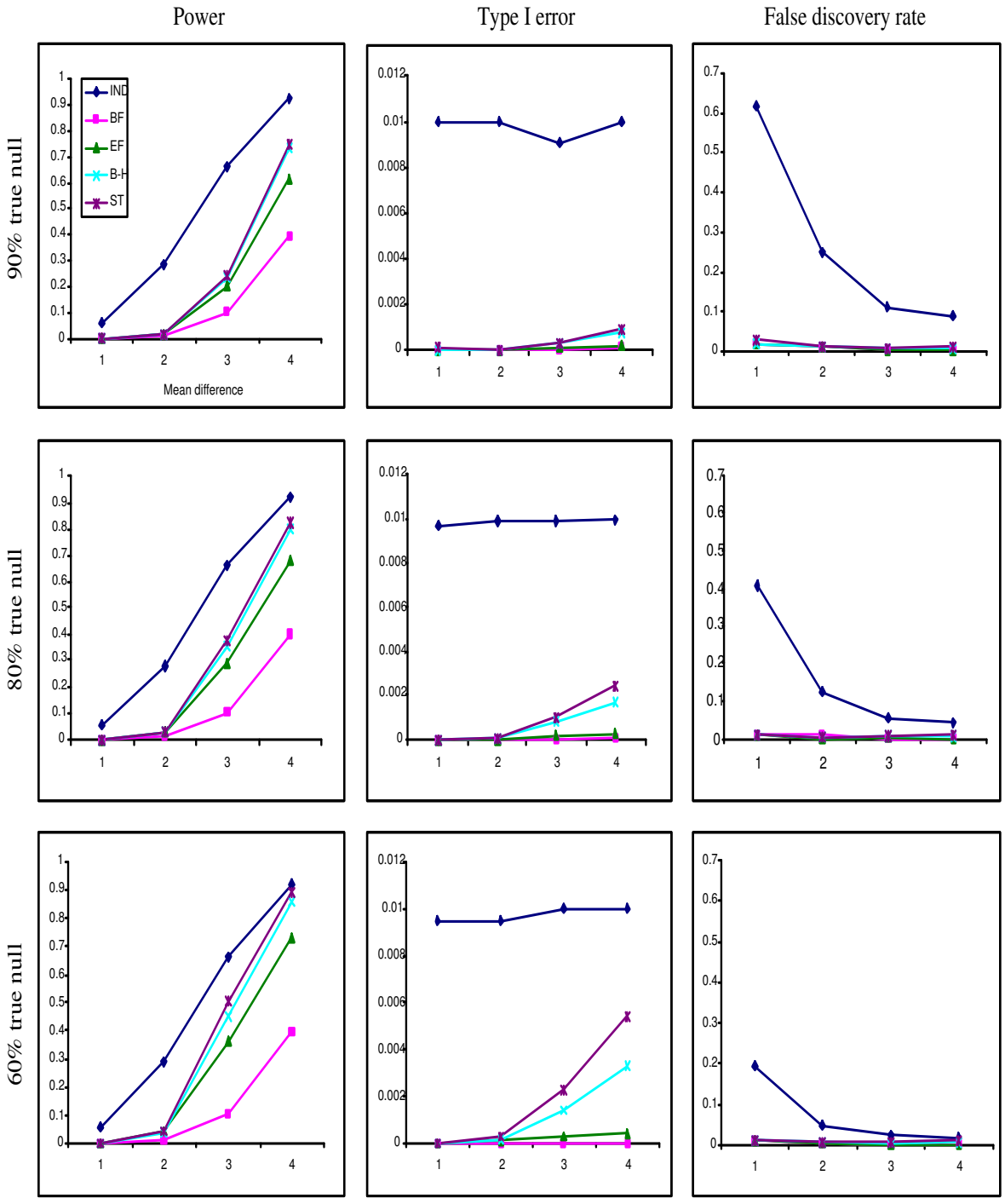

Figure 35: Average empirical power, type I error, and false discovery rates: individual test ($\diamond$), Bonferroni ($\square$), Efron ($\triangle$), Benjamini and Hochberg($\times$), the Storey ($*$).

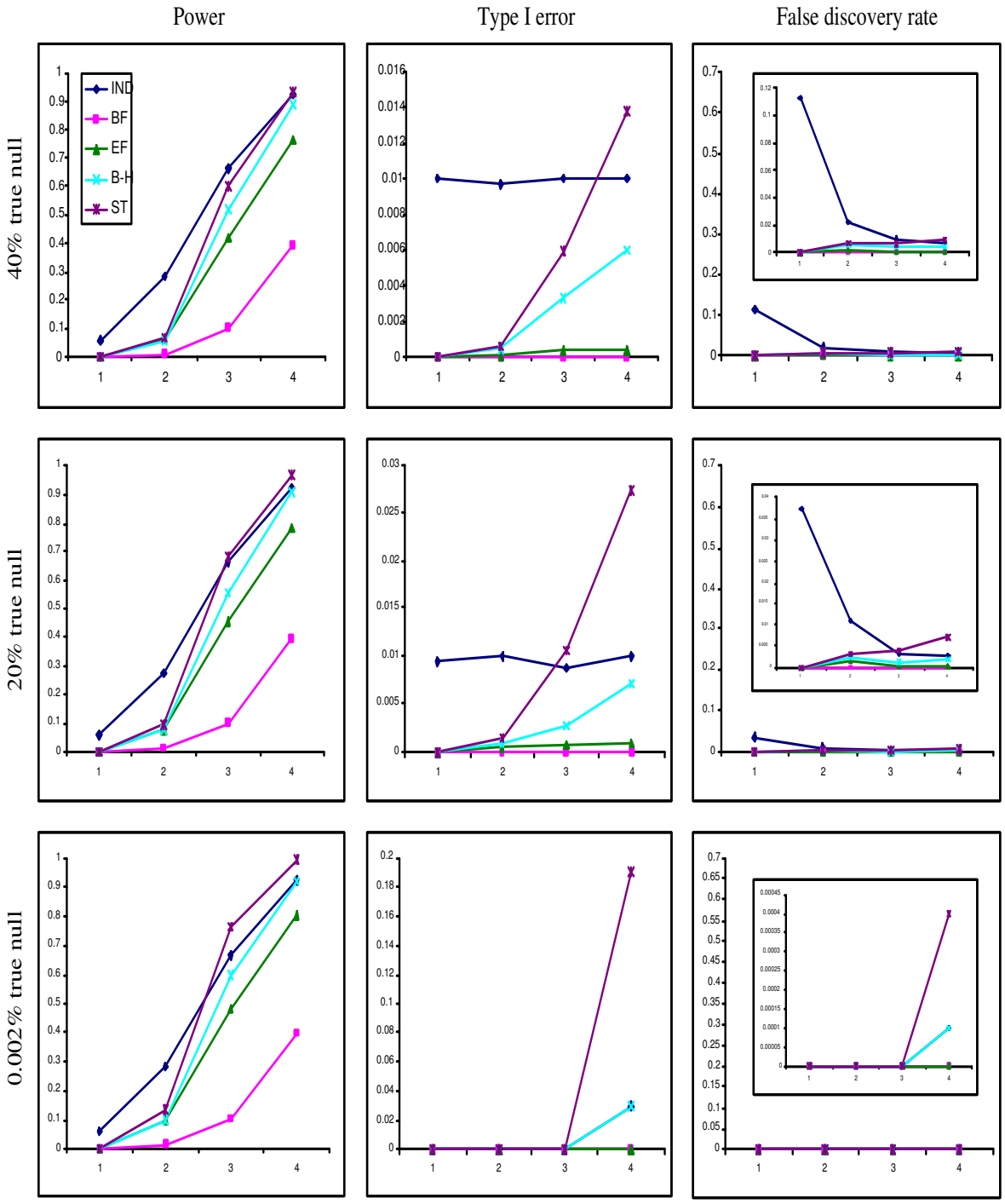

Figure 36: (Continuation of Figure 35) Average empirical power, type I error, and false discovery rates.

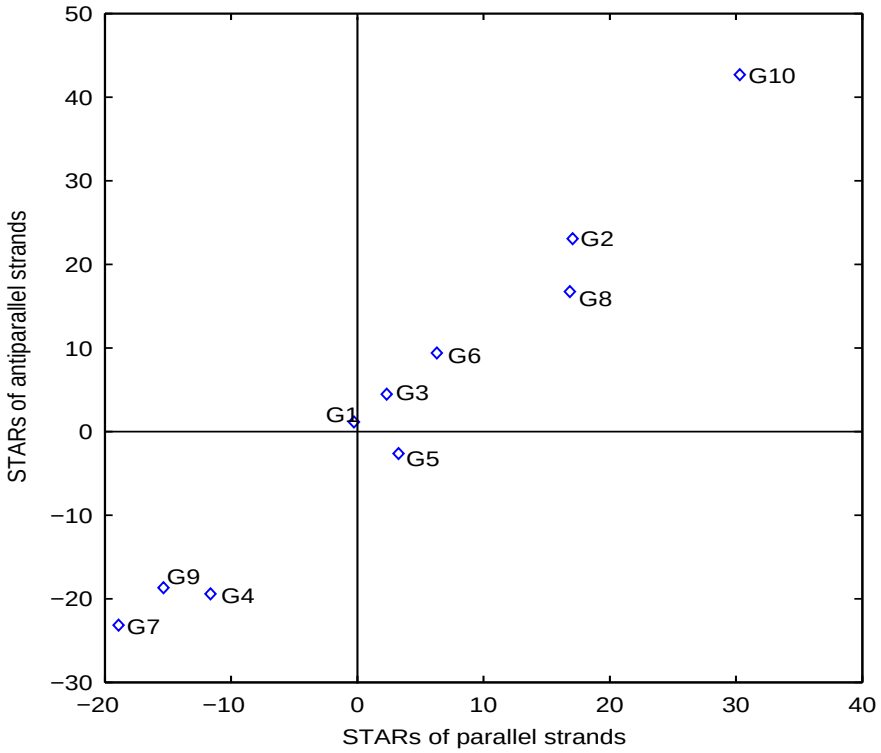

Figure 37: Associated pattern of grouped amino acid pairs between parallel and antiparallel strands. See Table 26 for definitions of G1 $\sim$ G10.

Table 29: Favored amino acid pairs in parallel β-sheet bridges. Five procedures (Individual (IND), Bonferroni (BF), Benjamini-Hochberg (B-H), Efron (EF), and Storey (ST)) for controlling corresponding false positive rates (FPR, FWER, FDR, Local FDR, and pFDR) at $\alpha = 0.01$ are applied to find significant pairs. S and N represent significant and nonsignificant pairs

Rank	Amino acid pair	STAR	p-value	Local FDR	IND	BF	B-H	EF	ST
1	E:K	16.43	0	0	S	S	S	S	S
2	D:R	13.65	0	0	S	S	S	S	S
3	N:T	11.86	0	0	S	S	S	S	S
4	I:I	10.07	0	0	S	S	S	S	S
5	D:K	10.04	0	0	S	S	S	S	S
6	I:L	8.39	0	0	S	S	S	S	S
7	V:V	7.95	0	0	S	S	S	S	S
8	N:N	7.65	0	0	S	S	S	S	S
9	H:T	6.68	0	0	S	S	S	S	S
10	T:T	6.48	0	0	S	S	S	S	S
11	K:T	6.38	0	0	S	S	S	S	S
12	E:R	6.31	0	0	S	S	S	S	S
13	S:T	6.24	0	0	S	S	S	S	S
14	R:T	6.11	0	0	S	S	S	S	S
15	Q:W	5.59	0	0	S	S	S	S	S
16	D:H	4.94	0	0	S	S	S	S	S
17	M:M	4.85	0	0	S	S	S	S	S
18	P:T	4.75	0	0.00013	S	S	S	S	S
19	Q:Y	4.72	0	0.00014	S	S	S	S	S
20	F:V	4.62	0	0.00023	S	S	S	S	S
21	N:Q	4.53	0	0.00032	S	S	S	S	S
22	S:S	4.34	0.00002	0.00069	S	S	S	S	S
23	E:H	4.18	0.00002	0.00130	S	S	S	S	S
24	M:P	3.95	0.00008	0.00311	S	N	S	S	S
25	Y:Y	3.80	0.00014	0.00529	S	N	S	S	S
26	C:R	3.76	0.00018	0.00607	S	N	S	S	S
27	D:S	3.70	0.00022	0.00724	S	N	S	S	S
28	L:V	3.35	0.00080	0.02227	S	N	S	N	S
29	L:L	3.08	0.00204	0.04857	S	N	S	N	S
30	C:V	3.08	0.00210	0.04946	S	N	S	N	S
31	I:V	2.96	0.00308	0.06818	S	N	S	N	S
32	A:L	2.94	0.00328	0.07137	S	N	N	N	S
33	K:P	2.90	0.00374	0.08026	S	N	N	N	S
34	N:S	2.89	0.00386	0.08224	S	N	N	N	S
35	H:S	2.70	0.00688	0.13102	S	N	N	N	S
36	G:Y	2.67	0.00758	0.14167	S	N	N	N	S
37	Q:S	2.64	0.00830	0.15229	S	N	N	N	S
38	M:W	2.62	0.00876	0.15947	S	N	N	N	S
39	N:W	2.59	0.00952	0.16998	S	N	N	N	S
40	Q:T	2.56	0.01056	0.18514	N	N	N	N	S
41	H:K	2.54	0.01114	0.19299	N	N	N	N	S
42	F:G	2.46	0.01394	0.23061	N	N	N	N	S
					N	N	N	N	N
					⋮	⋮	⋮	⋮	⋮

Table 30: Unfavored amino acid pairs in parallel β-sheet bridges. See the caption of Table 29 for definitions of columns

Rank	Amino acid pair	STAR	p-value	Local FDR	IND	BF	B-H	EF	ST
210	I:R	-7.25	0	0	S	S	S	S	S
209	I:S	-6.94	0	0	S	S	S	S	S
208	I:N	-6.36	0	0	S	S	S	S	S
207	N:V	-6.27	0	0	S	S	S	S	S
206	L:T	-5.85	0	0	S	S	S	S	S
205	T:V	-5.70	0	0	S	S	S	S	S
204	H:I	-5.31	0	0	S	S	S	S	S
203	I:K	-5.28	0	0	S	S	S	S	S
202	R:V	-5.24	0	0	S	S	S	S	S
201	D:I	-5.19	0	0	S	S	S	S	S
200	A:T	-5.07	0	0	S	S	S	S	S
199	I:T	-4.95	0	0	S	S	S	S	S
198	E:V	-4.92	0	0	S	S	S	S	S
197	G:I	-4.77	0	0.00010	S	S	S	S	S
196	L:N	-4.76	0	0.00011	S	S	S	S	S
195	E:L	-4.75	0	0.00011	S	S	S	S	S
194	I:Q	-4.69	0	0.00015	S	S	S	S	S
193	L:S	-4.65	0	0.00017	S	S	S	S	S
192	L:Q	-4.60	0	0.00022	S	S	S	S	S
191	D:V	-4.53	0	0.00029	S	S	S	S	S
190	K:V	-4.33	0.00002	0.00066	S	S	S	S	S
189	F:K	-4.31	0.00002	0.00071	S	S	S	S	S
188	L:W	-4.26	0.00002	0.00086	S	S	S	S	S
187	H:V	-4.24	0.00002	0.00096	S	S	S	S	S
186	A:M	-4.12	0.00004	0.00151	S	S	S	S	S
185	D:F	-4.05	0.00006	0.00198	S	N	S	S	S
184	K:L	-3.99	0.00006	0.00244	S	N	S	S	S
183	F:T	-3.95	0.00008	0.00287	S	N	S	S	S
182	S:V	-3.32	0.00089	0.02354	S	N	S	N	S
181	Q:V	-3.23	0.00126	0.03167	S	N	S	N	S
180	A:N	-3.19	0.00143	0.03539	S	N	S	N	S
179	D:W	-3.10	0.00197	0.04625	S	N	S	N	S
178	E:I	-3.05	0.00227	0.05226	S	N	S	N	S
177	V:Y	-3.05	0.00231	0.05310	S	N	S	N	S
176	W:Y	-3.04	0.00237	0.05409	S	N	S	N	S
175	D:Y	-2.82	0.00476	0.09669	S	N	S	N	S
174	C:Q	-2.72	0.00646	0.12438	S	N	S	N	S
173	T:W	-2.61	0.00911	0.16468	S	N	N	N	S
172	M:T	-2.50	0.01256	0.21350	N	N	N	N	S
171	C:T	-2.41	0.01590	0.25571	N	N	N	N	S
					N	N	N	N	N
					⋮	⋮	⋮	⋮	⋮

Table 31: Favored amino acid pairs in antiparallel β-sheet bridges. See the caption of Table 29 for definitions of columns

Rank	Amino acid pair	STAR	p-value	Local FDR	IND	BF	B-H	EF	ST
1	E:K	21.73	0	0	S	S	S	S	S
2	E:R	18.06	0	0	S	S	S	S	S
3	N:T	13.53	0	0	S	S	S	S	S
4	L:L	12.43	0	0	S	S	S	S	S
5	D:R	12.21	0	0	S	S	S	S	S
6	T:T	12.20	0	0	S	S	S	S	S
7	S:T	10.46	0	0	S	S	S	S	S
8	C:C	9.64	0	0	S	S	S	S	S
9	I:I	9.47	0	0	S	S	S	S	S
10	V:V	9.32	0	0	S	S	S	S	S
11	I:V	9.15	0	0	S	S	S	S	S
12	I:L	9.06	0	0	S	S	S	S	S
13	D:H	8.82	0	0	S	S	S	S	S
14	N:S	8.73	0	0	S	S	S	S	S
15	S:S	8.22	0	0	S	S	S	S	S
16	K:Q	7.48	0	0	S	S	S	S	S
17	K:T	7.18	0	0	S	S	S	S	S
18	Q:T	7.10	0	0	S	S	S	S	S
19	P:W	6.98	0	0	S	S	S	S	S
20	H:H	6.42	0	0	S	S	S	S	S
21	D:K	6.38	0	0	S	S	S	S	S
22	F:L	6.35	0	0	S	S	S	S	S
23	D:T	5.47	0	0	S	S	S	S	S
24	F:F	5.46	0	0	S	S	S	S	S
25	D:Q	5.43	0	0	S	S	S	S	S
26	A:A	5.18	0	0	S	S	S	S	S
27	K:Y	5.08	0	0	S	S	S	S	S
28	K:S	5.03	0	0	S	S	S	S	S
29	R:T	4.95	0	0	S	S	S	S	S
30	N:N	4.87	0	0	S	S	S	S	S
31	D:S	4.81	0	0	S	S	S	S	S
32	L:M	4.73	0	0.00012	S	S	S	S	S
33	F:Y	4.55	0	0.00028	S	S	S	S	S
34	G:S	4.35	0.00002	0.00064	S	S	S	S	S
35	A:I	4.31	0.00002	0.00076	S	S	S	S	S
36	K:N	4.24	0.00002	0.00100	S	S	S	S	S
37	C:I	4.21	0.00002	0.00115	S	S	S	S	S
38	D:D	3.92	0.00008	0.00347	S	N	S	S	S
39	E:Q	3.90	0.00010	0.00367	S	N	S	S	S
40	E:T	3.51	0.00046	0.01453	S	N	S	N	S
41	F:G	3.46	0.00054	0.01716	S	N	S	N	S
42	A:L	3.39	0.00070	0.02129	S	N	S	N	S
43	A:Y	3.38	0.00072	0.02178	S	N	S	N	S
44	M:M	3.37	0.00076	0.02259	S	N	S	N	S
45	Q:S	3.34	0.00084	0.02503	S	N	S	N	S
46	L:V	3.31	0.00094	0.02739	S	N	S	N	S
47	M:V	3.29	0.00098	0.02858	S	N	S	N	S
48	P:Y	3.27	0.00106	0.03044	S	N	S	N	S
49	C:F	3.19	0.00142	0.03902	S	N	S	N	S
50	E:H	3.12	0.00178	0.04767	S	N	S	N	S
51	G:W	3.00	0.00268	0.06789	S	N	S	N	S
52	H:N	2.97	0.00302	0.07509	S	N	S	N	S
53	A:F	2.84	0.00444	0.09851	S	N	S	N	S
54	F:V	2.72	0.00650	0.14423	S	N	N	N	S
55	F:I	2.61	0.00908	0.19062	S	N	N	N	S
56	A:V	2.51	0.01214	0.24308	N	N	N	N	S
57	C:W	2.43	0.01494	0.28851	N	N	N	N	S
58	G:Y	2.40	0.01640	0.31148	N	N	N	N	S
59	F:W	2.32	0.00206	0.37510	N	N	N	N	S
60	G:G	2.26	0.02394	0.42423	N	N	N	N	S
61	L:W	2.23	0.02582	0.45096	N	N	N	N	S
62	Q:Q	2.20	0.02752	0.47496	N	N	N	N	S
63	G:V	2.15	0.03164	0.53167	N	N	N	N	S
					N	N	N	N	N
					⋮	⋮	⋮	⋮	⋮

Table 32: Unfavored amino acid pairs in antiparallel β-sheet bridges. See the caption of Table 29 for definitions of columns

Rank	Amino acid pair	STAR	p-value	Local FDR	IND	BF	B-H	EF	ST
210	L:T	-12.48	0	0	S	S	S	S	S
209	S:V	-8.57	0	0	S	S	S	S	S
208	I:N	-8.24	0	0	S	S	S	S	S
207	I:T	-8.19	0	0	S	S	S	S	S
206	L:S	-8.11	0	0	S	S	S	S	S
205	D:V	-8.06	0	0	S	S	S	S	S
204	G:K	-7.53	0	0	S	S	S	S	S
203	E:I	-7.46	0	0	S	S	S	S	S
202	I:S	-7.45	0	0	S	S	S	S	S
201	F:T	-7.40	0	0	S	S	S	S	S
200	T:Y	-7.00	0	0	S	S	S	S	S
199	K:V	-6.99	0	0	S	S	S	S	S
198	D:I	-6.88	0	0	S	S	S	S	S
197	T:V	-6.81	0	0	S	S	S	S	S
196	F:K	-6.73	0	0	S	S	S	S	S
195	A:K	-6.68	0	0	S	S	S	S	S
194	E:F	-6.51	0	0	S	S	S	S	S
193	I:K	-6.49	0	0	S	S	S	S	S
192	T:W	-6.39	0	0	S	S	S	S	S
191	F:N	-6.34	0	0	S	S	S	S	S
190	L:Q	-6.18	0	0	S	S	S	S	S
189	S:Y	-6.06	0	0	S	S	S	S	S
188	K:L	-6.03	0	0	S	S	S	S	S
187	D:L	-5.95	0	0	S	S	S	S	S
186	N:V	-5.75	0	0	S	S	S	S	S
185	L:N	-5.64	0	0	S	S	S	S	S
184	I:P	-5.62	0	0	S	S	S	S	S
183	E:L	-5.54	0	0	S	S	S	S	S
182	L:R	-5.44	0	0	S	S	S	S	S
181	D:F	-5.31	0	0	S	S	S	S	S
180	A:Q	-4.90	0	0	S	S	S	S	S
179	E:G	-4.83	0	0	S	S	S	S	S
178	I:Q	-4.68	0	0.00013	S	S	S	S	S
177	A:E	-4.67	0	0.00014	S	S	S	S	S
176	F:S	-4.33	0.00002	0.00062	S	S	S	S	S
175	D:W	-4.19	0.00002	0.00108	S	S	S	S	S
174	D:Y	-4.19	0.00002	0.00108	S	S	S	S	S
173	F:R	-4.14	0.00004	0.00137	S	S	S	S	S
172	R:V	-4.06	0.00004	0.00184	S	S	S	S	S
171	C:S	-4.02	0.00006	0.00212	S	N	S	S	S
170	C:T	-3.97	0.00008	0.00258	S	N	S	S	S
169	K:R	-3.95	0.00008	0.00288	S	N	S	S	S
168	C:Q	-3.93	0.00008	0.00308	S	N	S	S	S
167	H:L	-3.87	0.00010	0.00376	S	N	S	S	S
166	Q:V	-3.81	0.00014	0.00472	S	N	S	S	S
165	H:W	-3.70	0.00022	0.00716	S	N	S	S	S
164	G:R	-3.70	0.00022	0.00721	S	N	S	S	S
163	M:T	-3.68	0.00024	0.00760	S	N	S	S	S
162	C:E	-3.53	0.00040	0.01276	S	N	S	N	S
161	E:P	-3.45	0.00056	0.01710	S	N	S	N	S
160	G:T	-3.42	0.00062	0.01846	S	N	S	N	S
159	A:R	-3.33	0.00086	0.02493	S	N	S	N	S
158	I:R	-3.33	0.00088	0.02522	S	N	S	N	S
157	M:N	-3.32	0.00090	0.02611	S	N	S	N	S
156	H:V	-3.31	0.00092	0.02654	S	N	S	N	S
155	E:W	-3.24	0.00120	0.03318	S	N	S	N	S
154	E:V	-3.20	0.00138	0.03791	S	N	S	N	S
153	E:Y	-2.93	0.00334	0.08234	S	N	S	N	S
152	A:N	-2.93	0.00336	0.08274	S	N	S	N	S
151	N:Y	-2.86	0.00428	0.09234	S	N	S	N	S
150	A:T	-2.75	0.00588	0.13475	S	N	N	N	S
149	A:D	-2.63	0.00856	0.18603	S	N	N	N	S
148	F:Q	-2.61	0.00906	0.19529	S	N	N	N	S
147	M:Q	-2.45	0.01420	0.28603	N	N	N	N	S
146	A:C	-2.33	0.01954	0.37327	N	N	N	N	S
145	C:D	-2.24	0.02484	0.45520	N	N	N	N	S
144	G:H	-2.23	0.02574	0.46829	N	N	N	N	S
					N	N	N	N	N
					⋮	⋮	⋮	⋮	⋮

Table 33: Summary of multiple testing results

		# of sig. favored pairs		# of sig. unfavored pairs	
Procedure	False positive rate	Parallel	Antiparallel	Parallel	Antiparallel
Individual test	Indiviaul=0.01, FWER=0.88	39	55	38	63
Bonferroni	Individual=0.000048, FWER=0.01	23	37	25	39
Benjamin-Hochberg	FDR=0.01	31	53	35	60
Efron	Threshold of Local FDR=0.01	27	39	28	48
Storey	pFDR=0.01	42	63	40	67

• Interpretation of Patterns of Individual Amino Acid Pairs Using Information Gain

Once we identified the significant pairs by the multiple testing procedure, the next step is to interpret the results biologically. We utilize the chemical properties of amino acids based on the four groups described in Section 4.5.3. Furthermore, we employ information gain (IG) for identifying the amino acid groups that classify favored and unfavored pairs. IG has been implemented in C4.5 popular tree-based model to select important variables. More formally, IG is the expected reduction in entropy caused by partitioning the data according to the variable (Mitchell, 1997). Hence, the variable having highest information gain provides the best classification rate of the response variable. More precisely, the IG of a specific variable ($\mathcal{V}$) given data ($\mathcal{D}$) is defined as follows:

$$\text{IG}(\mathcal{V}, \mathcal{D}) \equiv \text{Entropy } (\mathcal{D}) - \sum_{k \in value(\mathcal{V})} \frac{\#\mathcal{D}_k}{\#\mathcal{D}} \cdot \text{Entropy}(\mathcal{D}_k), \tag{35}$$

where Entropy ($\mathcal{D}$) is

$$\text{Entropy}(\mathcal{D}) \equiv \sum_{j=1}^{C} -p_i \log_2 p_i, \tag{36}$$

where C is the number of classes and p_i is the proportion of $\mathcal{D}$ belonging to class j. $value(\mathcal{V})$ is all possible values of variable $\mathcal{V}$ and $\mathcal{D}_k$ is the subset of $\mathcal{D}$ in which variable $\mathcal{V}$ has value k. To illustrate the computing process of information gain in our example, let's construct the data set containing the variables and classes of each pair in parallel strands (Table 34) and in antiparallel strands (Table 35). Each variable (i.e., grouped amino acid) can take on the value 1 or 0. The value 1 indicates that an individual pair is in this group whereas the value 0 indicates an individual pair is not in this group. The two classes depend on whether an amino acid pair is favored or unfavored. As an example, information gain of a Group 10 (G10) in parallel strands

can be computed as follows:

$$\begin{aligned} value(\text{G10}) &= 1, 0 \\ \mathcal{D} &= [43F, 39U] \\ \mathcal{D}_1 &= [13F, 3U] \\ \mathcal{D}_0 &= [29F, 36U] \end{aligned}$$

where F=Favored and U=Unfavored.

$$\text{IG(G10, } \mathcal{D}) = \text{Entropy } (\mathcal{D}) - \sum_{k \in value\{1,0\}} \frac{\#\mathcal{D}_k}{81} \cdot \text{Entropy}(\mathcal{D}_k),$$

$$\begin{aligned} \text{Entropy}(\mathcal{D}) &= -\frac{42}{81}\log_2\left(\frac{42}{81}\right) - \frac{39}{81}\log_2\left(\frac{39}{81}\right) = 0.99901, \\ \text{Entropy}(\mathcal{D}_1) &= -\frac{13}{16}\log_2\left(\frac{13}{16}\right) - \frac{3}{16}\log_2\left(\frac{3}{16}\right) = 0.13752, \\ \text{Entropy}(\mathcal{D}_0) &= -\frac{29}{65}\log_2\left(\frac{29}{65}\right) - \frac{36}{65}\log_2\left(\frac{36}{65}\right) = 0.79574. \end{aligned}$$

Thus, IG(G10, $\mathcal{D}$) = 0.0657.

Table 36 contains the completed results of information gain for all groups in parallel and antiparallel strands. The groups with high information gain can be considered as the significant variable to the classification between favored and unfavored pairs. Here, we use the threshold to determine significant pairs at 0.02, which can be subjective. Interesting observation here is that parallel and antiparallel strands identify the significant features with different order. Let's elaborate on these observations in parallel and antiparallel strands, respectively.

Parallel Strands: With regard to favored pairs, G8, G2, G10, and G6 provide high information gain in decreasing order. G8 involves the association within uncharged polar amino acids. G2 involves the association between negatively charged

Table 34: Portion of data set containing the features and classes of each pair in parallel strands and summary

Index	Pairs	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	Class
1	E:K	0	1	0	0	0	0	0	0	0	0	Favored
2	D:R	0	1	0	0	0	0	0	0	0	0	Favored
...	...	...	...	...	...	...	...	...	...	...	...	...
41	H:K	0	0	0	0	1	0	0	0	0	0	Favored
42	F:G	0	0	0	0	0	0	0	0	0	1	Favored
43	I:R	0	0	0	0	0	0	1	0	0	0	Unfavored
44	I:S	0	0	1	0	0	0	0	0	1	0	Unfavored
...	...	...	...	...	...	...	...	...	...	...	...	...
80	T:W	0	0	0	0	0	0	0	0	1	0	Unfavored
81	M:T	0	0	0	0	0	0	0	0	1	0	Unfavored

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	Total
Favored	0	6	1	1	1	4	1	11	4	13	42
Unfavored	0	0	1	7	0	0	8	0	20	3	39
Total	0	6	2	8	1	4	9	11	24	16	81

polar and positively charged polar amino acids. G10 represents the association within polar amino acids. Finally, G6 involves the association between positively charged polar and uncharged polar amino acids. All individual pairs that exhibit the above described patterns can be found in Table 29. Note that the results in this Section provide more precise information than the results in Section 4.5.4 (Pattern recognition of grouped amino acid pairs).

Observations on pairs which exhibit a significantly unfavored pattern in parallel strands are presented in Table 30. Such pairs are rarely observed to form β-sheet bridges. According to information gain, G9, G7, and G4 provide high information gain in decreasing order. This implies that nonpolar amino acids tend to not be associated with uncharged polar amino acids. Moreover, both positively and negatively charged polar amino acids show no inclination to interact with nonpolar amino acids.

Antiparallel Strands: Tables 31 and 32 give the lists of significantly favored

Table 35: Portion of data set containing the features and classes of each pair in antiparallel strands and summary

Index	Pairs	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	Class
1	E:K	0	1	0	0	0	0	0	0	0	0	Favored
2	E:R	0	1	0	0	0	0	0	0	0	0	Favored
...	...	...	...	...	...	...	...	...	...	...	...	...
62	Q:Q	0	0	0	0	0	0	0	1	0	0	Favored
63	G:V	0	0	0	0	0	0	0	0	0	1	Favored
64	L:T	0	0	0	1	0	0	0	0	1	0	Unfavored
65	S:V	0	0	0	0	0	0	0	0	1	0	Unfavored
...	...	...	...	...	...	...	...	...	...	...	...	...
129	C:D	0	0	0	1	0	0	0	0	0	0	Unfavored
130	G:H	0	0	0	0	0	0	1	0	0	0	Unfavored

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	Total
Favored	1	6	5	0	1	7	0	9	5	29	63
Unfavored	0	0	2	16	1	0	16	3	27	2	67
Total	1	6	7	16	2	7	16	12	32	31	130

Table 36: Information gains of the features in parallel and antiparallel strands

Parallel Strands			Antiparallel Strands		
Index	IG	Remark	Index	IG	Remark
G9	0.1602	Unfavored	G10	0.2103	Favored
G8	0.1430	Favored	G4	0.1294	Unfavored
G2	0.0742	Favored	G7	0.1294	Unfavored
G7	0.0666	Unfavored	G9	0.1100	Unfavored
G10	0.0657	Favored	G6	0.0586	Favored
G4	0.0540	Unfavored	G2	0.0500	Favored
G6	0.0485	Favored	G8	0.0214	Favored
G5	0.0118	N/A	G3	0.0089	N/A
G3	0.00002	N/A	G1	0.0081	N/A
G1	0	N/A	G5	0.00001	N/A

and unfavored pairs in antiparallel strands. Overall patterns of antiparallel strands are similar to those in parallel strands. However, as shown in Table 36, the order of significant variables is different from that of parallel strands. For favored pairs in

antiparallel strands, G10, G6, G2, and G8 provide high information gain in decreasing order, whereas parallel strands provide information gain with different order (i.e., G8, G2, G10, and G6). For unfavored pairs, the order of significant variable is completely opposite between parallel and antiparallel strands. In antiparallel strands, the order of significant features is G4, G7, and G9 but parallel strands order the significant variables the other way around. This discrepancy of patterns between parallel and antiparallel strands motivates further analysis, as illustrated in the following Section.

4.5.5 Graphical Analysis of Discrepancy Between Parallel and Antiparallel Strands

Figure 38 is the scatter plot of the STAR values computed from all (210) individual amino acid pairs in parallel and antiparallel strands. The STAR values can be categorized into four groups based on their sign. The small box inside Figure 38 is the boundary of the significant pairs identified by the Storey's procedure. Thus, the pairs outside of the box are significant for at least one of the strands. Table 37 gives the list of pairs in each group where the pairs in the parentheses indicate significance for at least one of the strands.

1. First group (I) : The STAR values of the amino acid pairs in both parallel and antiparallel strands are positive (displayed in first quadrant). 58 pairs out of 81 are significant.

2. Second group (II): The STAR values of the amino acid pairs in parallel strands are negative but those in antiparallel strands are positive (displayed in second quadrant). 13 pairs out of 25 are significant.

3. Third group (III): The STAR values of the amino acid pairs in both parallel and antiparallel strands are negative (displayed in third quadrant). 59 pairs out of 75 are significant.

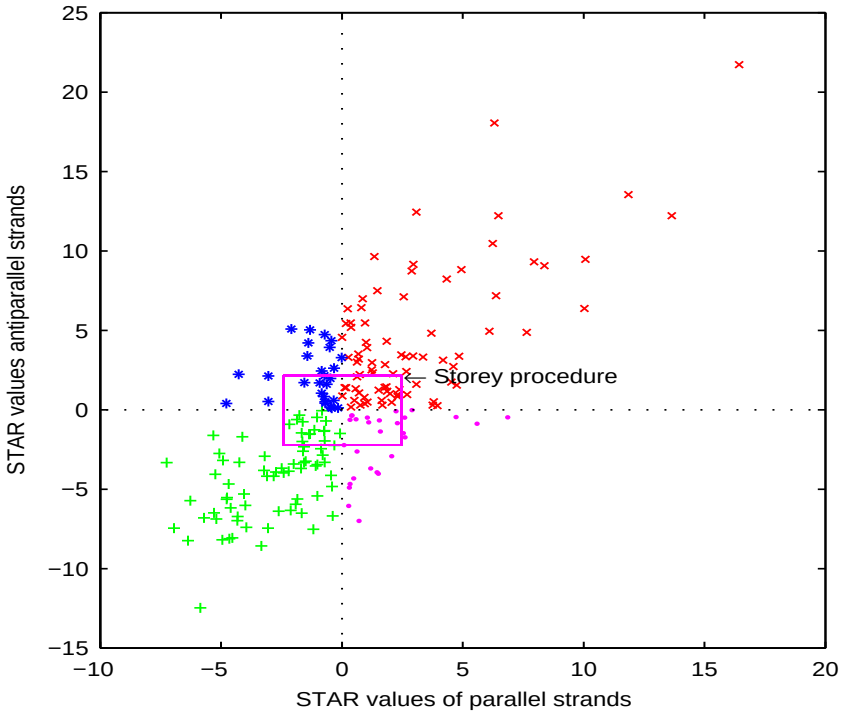

Figure 38: Associated pattern of individual amino acid pairs between parallel and antiparallel strands.

4. Fourth group (IV): The STAR values of the amino acid pairs in parallel strands are positive but those in antiparallel strands are negative (displayed in fourth quadrant). 18 pairs out of 29 significant.

The amino acid pairs in the first group have favored propensity for pair-wise associations in both parallel and antiparallel strands. On the other hand, the amino acid pairs in the third group show unfavored tendency for the association in both strands. The amino acid pairs with the opposite propensity for the association between parallel and antiparallel strands are in the second and fourth groups, which are of interest. However, as Figure 38 shows, most of the significant pairs are in the first and third groups and only some pairs are significant in the second and fourth groups. Thus, we can cautiously conclude that the patterns of association of pair-wise amino acids in

β-sheet bridges between parallel and antiparallel strands are not significantly different. However, it is still interesting to carry out further investigation of the individual amino acids with different patterns.

4.5.6 Verification of Asymptotic Normality of STAR

In this section we revisit the STAR values of individual amino acid pairs shown in Tables 29 $\sim$ 32. The STAR values were calculated for each cell, i.e., $\tilde{e}_{11}, \tilde{e}_{12}, \cdots, \tilde{e}_{rc}$. And they are assumed to be independent. We know that STAR values follow a standard normal distribution in an asymptotic sense. Hence, we compute the p-values in a general way. However, since this asymptotic property is not valid for every case, we compute the p-values by a resampling method for evaluation of asymptotic assumption. Consider random sampling of two amino acids among the database and make them a pair. Under the assumption of independence of each pair, the STAR values are computed for r $\times$ c pairs. If this procedure is repeated $\mathcal{N}$ times, we can get a set of null statistics $\tilde{e}_{11}^{(n)}, \tilde{e}_{12}^{(n)}, \cdots, \tilde{e}_{rc}^{(n)}$, $n = 1, 2, \cdots, \mathcal{N}$. Then the empirical p-value for cell ij, where i=$1, 2, \cdots$, r and j=$1, 2, \cdots$, c is calculated as

$$p_{ij} = \sum_{n=1}^{\mathcal{N}} \frac{\#\{|\tilde{e}_{ij}^{(n)}| \geq |\tilde{e_{ij}}|\}}{\mathcal{N}}. \tag{37}$$

The p-values obtained from the resampling method are compared with the p-values obtained from the assumption of asymptotical normality. We perform a paired t-test between two sets of p-values. A 95% confidence interval for mean difference of p-values from a resampling method and the asymptotically normal distribution is $(-0.00213, 0.000786)$, which include zero. Since the difference of p-values between the two approaches is not statistically significant, the asymptotically standard normal assumption is valid in our experiment.

Table 37: Amino acid pairs, grouped by the sign of STAR values between parallel and antiparallel strands

I		II	III		IV
(AA)	GN	(AY)	(AC)	(GR)	(AD)
(AF)	GQ	CH	AH	(GT)	(AE)
AG	(GW)	(CI)	(AK)	(HI)	(AQ)
(AI)	(GY)	(CW)	(AM)	(HL)	AS
(AL)	(HH)	(DD)	(AN)	HM	(CS)
(AV)	(HN)	FH	AP	(HV)	(EY)
AW	HQ	(GI)	(AR)	(IK)	FP
(CC)	(HS)	(GS)	(AT)	(IN)	(FS)
(CF)	HY	GV	(CD)	(IP)	(GH)
CL	(II)	HP	(CE)	(IQ)	GM
CM	(IL)	IY	CG	(IR)	(HK)
CN	IM	(KS)	CK	(IS)	HR
CP	(IV)	(KY)	(CQ)	(IT)	(HT)
(CR)	(KN)	(LM)	(CT)	(KL)	(HW)
(CV)	(KQ)	LP	CY	KM	IW
DE	(KT)	(LW)	(DF)	(KV)	KK
DG	KW	MS	(DI)	(LN)	(KP)
(DH)	(LL)	NP	(DL)	(LQ)	(KR)
(DK)	(LV)	PR	DM	(LR)	(MW)
DN	LY	(PY)	(DV)	(LS)	(NW)
DP	(MM)	RW	(DW)	(LT)	PQ
(DQ)	(MP)	VW	(DY)	(MN)	PS
(DR)	(MV)	(VY)	EE	(MQ)	(QW)
(DS)	(NN)	WW	(EF)	MR	(QY)
(DT)	(NQ)	(WY)	(EG)	(MT)	RR
(EH)	NR		(EI)	MY	RS
(EK)	(NS)		(EL)	(NV)	SW
EN	(NT)		EM	(NY)	(SY)
(EQ)	(PT)		(EP)	PP	(TY)
(ER)	(PW)		(EV)	PV	
ES	(QQ)		(EW)	(QV)	
(ET)	QR		(FI)	(RV)	
(FF)	(QS)		(FK)	(SV)	
(FG)	(QT)		(FN)	(TV)	
(FL)	(RT)		(FQ)	(TW)	
FM	RY		(FR)		
(FV)	(SS)		(FT)		
(FW)	(ST)		(GK)		
(FY)	(TT)		GL		
(GG)	(VV)		GP		
	(YY)				
81(58)		25(13)	75(59)		29(18)

4.6. Conclusions

A multiple testing procedure has been proposed for an inference of independence of categories in each cell in the contingency tables. This procedure compensates for the limitation of the globally significant tests such as χ^2 and likelihood ratio tests in that it provides more information about the nature of the association in each cell in the contingency tables. Moreover, the procedure has an advantage over the subjective methods such as normal probability plotting and partitioning of χ^2. In large-scale contingency tables in particular, the proposed procedure provides an objective and systematic way of finding the significantly associated cells. In this Chapter, four multiple testing procedures (Bonferroni, Benjamini-Hochberg, Efron, and Storey) that control corresponding compound errors (FWER, FDR, Local FDR, pFDR) are utilized and compared. The simulation studies show that the procedures controlling pFDR, FDR, and Local FDR provide higher power than BF method controlling classical FWER. The high power allows further characterization of the identified cells. For the case study, the proposed procedure has been applied to identify the patterns of pair-wise association of amino acids in β-sheet bridges and produced a list of favored and unfavored pairs. The statistical procedure considered in this Chapter cannot identify the physical or chemical nature of observed associations. However, these results are useful for better understanding of protein structure and should help develop better algorithms of the protein secondary and tertiary structure prediction.

CHAPTER V

CONCLUSION

Discovering meaningful rules and patterns from complex data set has been one of the majors tasks in a wide range of fields. Even though many commercial software packages are available, there is still much room to improve current techniques as well as develop new ideas to deal with emerging problems such as biomedical and environmental studies. Among the many opportunities in data mining, this thesis investigates the tree-based models and multiple testing in large-scale contingency tables. We propose a frontier-based tree-pruning algorithm (FBP), which provides a graphical way to implement the task of minimizing the complexity penalized loss function. Compared to the cost-complexity pruning method, FBP provides a full spectrum of information in tree pruning including an automatic identification of inadmissible tree sizes. A combination of FBP and cross validation (CV) produces "better" classifiers in simulations, compared to other existing methods. Various simulation studies show the justification of the FBP method. Results indicate that the number of admissible tree sizes is always a small proportion (roughly 10%) of all possible tree sizes. Also FBP always gives smaller CV error and tree size than CCP. With regards to testing errors, FBP and CCP are comparable. For an application, FBP can be utilized to study the stability of applying CV in building tree models.

As an extension of tree-based modeling, we study the behavior of the tree-based classifier selected by the CV method using simulations. This study leads to the following observation: The difference between testing and training errors from a cross-validated tree classifier is comparable to that of an oracle classifier, within a constant

factor. Various simulations justify the results. slope and R^2 are employed as a measure of the degree of relationship. Both the slope and R^2 being equal to 1 suggest a strong relationship between two classifiers. Additionally, we demonstrate that the above relationship is influenced by other factors such as the geometry of decision boundaries, the probabilistic parameter of an underlying distribution, and the sample sizes. There are two interesting directions for future research. One is theoretical and the other is numerical. On the theoretical front, all the conjectures are waiting to be proven. For more numerical studies, one can extend our study to other data mining algorithms, such as support vector machines, neural networks, and so on. Also our study can be used to identify a statistical relationship between the sizes of training and testing samples. This relationship can surely help improve classification accuracy.

In the second part of the thesis, we propose to apply the multiple testing procedure to statistical inference of independence of categories in each cell of a contingency table. The primary motivation of this study is to develop the follow-up testing method for χ^2 and likelihood ratio tests in contingency tables. The proposed procedure has an advantage over the other follow-up testing methods such as normal probability plotting and partitioning of χ^2. In large-scale contingency tables, in particular, the proposed procedure provides an objective and systematic way of identifying patterns of individual cells. We perform simulation studies to compare the power, type I error, and false discovery rate of five testing procedures (i.e., the individual test, Bonferroni, Benjamini-Hochberg, Efron, and Storey procedures) that control corresponding false positive rates. To further understand the behavior of testing procedures, we perform simulations with a normal random variable. The simulation results show that the individual test procedure renders large power but it produces larger type I errors and false discovery rates than the others. If the proportion of true null hypotheses decreases and δ (defined in Section 4.4.3) increases, Storey's procedure gives large

power with relatively small type I error and false discovery rates.

The multiple testing procedure in the contingency tables is demonstrated by identifying the patterns of pair-wise associations of amino acids involved in beta-sheet bridges. The frequency of pair-wise amino acids in parallel and antiparallel strands is obtained from the 613 known proteins in the Protein Data Bank (PDB). After classifying each of 20 amino acids into four groups based on physicochemical properties, we found a number of patterns of individual residue pairs. Overall patterns indicate that pairs with the same chemical property (e.g., electrostatic, hydrophobic, or polar interactions) tend to interact, but the pairs in different groups do not. These patterns lead us to develop a better algorithm for the prediction of the secondary and tertiary structure of proteins.

APPENDIX A

DESCRIPTION OF DATA SETS FOR CHAPTER I

The 12 data sets are from the UCI repository.

1. **Wisconsin Breast Cancer.** This beast cancer data was given to the UCI repository by William H. Wolberg, University of Wisconsin Hospitals, Madison (see Wolberg and Mangasarian (1990)). The original dataset contains 699 cases, with 9 numeric attributes, and 2 classes (benign or malignant). The dataset we used contains 683 cases after removing cases with missing values. (subdirectory/breast-cancer-wisconsin/)

2. **Cleveland Heart Disease.** This is heart disease data gathered by Robert Detrano, V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. We use the preprocessed data in Statlog Project, which consists of 270 cases, with seven numeric and six categorical attributes, and two classes (absence or presence of heart disease). There are no missing values. (subdirectory/statlog/heart/)

3. **Pima Indians Diabetes.** This diabetes data was collected at National Institute of Diabetes and Digestive and Kidney Diseases, and donated to the UCI repository by Vincent Sigillito. It contains 768 cases, with 7 numeric attributes, and 2 classes (tested positive or negative for diabetes). No missing values. (subdirectory/pima-indians-diabetes/)

4. **BUPA Liver Disorders.** This UCI data was collected by Richard S. Forsyth, BUPA Medical Research Ltd. The dataset has 345 cases, with 6 numeric attributes, and 2 classes (a male patient with liver disorder or not). No missing values. (subdirectory/liver-disorders/).

5. **Congressional Voting Records.** Jeff Schlimmer donated the 1984 United Stated Congressional Voting Records to the UCI repository. This dataset consists of 435 cases, with 16 categorical attributes, and 2 classes (democrats or republicans). Missing attribute values are denoted by "?". (subdirectory/ voting-records/)

6. **Australian Credit Approval.** This credit data was initially used by Quinlan (1987) and Quinlan (1992). The preprocessed data in Statlog Project used for this study has 690 cases, with 6 numeric and 8 categorical attributes, and 2 classes (good or bad). There are 37 cases with one or more missing attribute values, which are replaced by the corresponding mode (categorical) or mean (numeric). (subdirectory/statlog/australian/)

7. **German Credit.** The German credit data was provided by Hans Hofmann. It contains 1000 cases, with 7 numeric and 13 categorical attributes, and 2 classes (good or bad). No missing values. (subdirectory/statlog/german/)

8. **Image Segmentation.** This image-pattern-recognition data was provided by Vision Group, University of Massachusetts. The samples were drawn randomly from a database of 7 outdoor images (brickface, sky, foliage, cement, window, path, grass). There are 7 classes, 19 numeric attributes and 2310 records in the dataset. (subdirectory/statlog/segment/)

9. **Vehicle Silhouettes.** This data originated from the Turing Institute, Glasgow, Scotland. The purpose is to classify a given silhouette as one of the four types of

vehicle: bus, van, saab, and opel. The dataset has 846 records, with 18 numeric attributes. (subdirectory/statlog/vehicle/)

10. **Satellite Image.** This data contains the multi-spectral values of pixels within 3x3 neighborhoods in a satellite image, and the classification associated with the central pixel in each neighborhood. The aim is to predict this classification given the multi-spectral values. There are 6 classes (red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, and very damp grey soil), and 36 numeric attributes. The training set has 4435 records, while the test set has 2000 records. (subdirectory /statlog/satimage)

11. **Iris Plants.** Created by R. A. Fisher, this is perhaps the most well-known dataset in the pattern recognition literature. It contains 3 classes of 50 records each, where each class refers to a type of iris plant (Setosa, Versicolour, or Virginica). There are 4 numeric attributes, and no missing values in the dataset. (subdirectory /iris)

12. **Waveform.** This is an artificial 3-class (with equal probabilities) problem. A description is given in Breiman *et al.* (2001), and a C program for generating the data is available from the UCI repository. We use the data generated by Lim *et al.* (2000). There are 21 numeric attributes, 600 records in the training set, and 3000 in the test set. (subdirectory/waveform)

APPENDIX B

ESTIMATION OF PFDR FOR CHAPTER III

Estimation of the q-value begins with estimating the pFDR. For multiple hypothesis problems, if the p-value is less than or equal to t, where $0 < t < 1$, we call the corresponding hypothesis "significant." Suppose there are m p-values, where m is the number of hypotheses (or number of cells in a contingency table): P_1, P_2, $\cdots$, P_m. Then the ordered p-values are expressed as $P_{(1)}$, $P_{(2)}$,$\cdots$,$P_{(m)}$,

$$\text{pFDR(t)} = \text{E}\left[\frac{V(t)}{R(t)}|R(t) > 0\right] \approx \frac{E[V(t)]}{E[R(t)]}; \quad \text{approximation holds if } m \text{ becomes large.}$$

$E[R(t)]$ and $E[V(t)]$ are estimated as follows.

$$\begin{aligned} \hat{E[R(t)]} &= \sum_{i=1}^{m} I(P_i \leq t), \\ \hat{E[V(t)]} &= \hat{\pi_0} mt \quad \text{where} \quad \hat{\pi_0} = \frac{m_0}{m}, \end{aligned}$$

$\hat{\pi_0}$, the proportion of the true null hypotheses, is estimated as follows:

$$\hat{\pi}_0(\lambda) = \frac{\sum_{i=1}^{m} I(P_i > \lambda)}{m(1-\lambda)}.$$

Optimal λ ($=\lambda_{\text{opt}}$) is determined to balance the bias variance trade off. Generally, when λ gets smaller, the bias term increases, but the variance decreases (See Storey and Tibshirani, 2003 for details):

$$\lambda_{\text{opt}} = \underset{\lambda}{\text{argmin}}[\text{Bias}^2\{\hat{\pi}_0(\lambda)\} + \text{Var}\{\hat{\pi}_0(\lambda)\}].$$

Now the estimate of the pFDR can be expressed as

$$\hat{\text{pFDR}}_\lambda(t) = \frac{\hat{\pi_0}(\lambda) mt}{\sum_{i=1}^{m} I(P_i \leq t)}.$$

APPENDIX C

EXAMPLE OF BENJAMINI AND HOCHBERG PROCEDURE FOR CHAPTER III

Recall that the Benjamini and Hochberg's proceure is: For fixed α, where $0 \leq \alpha \leq 1$, we find the $\hat{i}$, which maximize $\left[i : P_{(i)} \leq \frac{i}{m} \cdot \alpha\right]$, where m is the number of hypotheses, i is an index. Then following rejection rule can be made:

$$\Omega \in \{\text{reject all } \ H_i \ \text{ with } \ P_i \leq P_{(i)}\}.$$

This rejection rule achieves $\text{FDR}(\Omega) \leq \alpha$. Suppose we wish to test 15 null hypotheses $(H_1, \cdots, H_{15})$ simultaneously on the basis of independent test statistics $Y_1, Y_2, \cdots Y_{15}$. From these statistics, we compute corresponding p-values, $p_1, p_2, \cdots, p_{15}$ and ordered p-values, $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(15)}$. Table 38 explains how this procedure works. Note that after the fourth ordered p-value, p-value becomes larger than $\frac{i}{m} \cdot \alpha$. Thus all hypotheses whose p-value is smaller than fourth p-value are significant.

Table 38: Example of Bejamini and Hochberg's multiple testing procedure at $\alpha = 0.01$

Rank (i)	p-value	$\frac{i}{m} \cdot \alpha$	significance
1	0.0001	0.0033	Significant (S)
2	0.0004	0.0067	S
3	0.0019	0.0100	S
4	0.0095	0.0133	S
5	0.0201	0.0167	Not Significant (NS)
6	0.0278	0.0200	NS
7	0.0298	0.0233	NS
8	0.0344	0.0267	NS
9	0.0495	0.0300	NS
10	0.3240	0.0333	NS
11	0.4262	0.0367	NS
12	0.5719	0.0400	NS
13	0.6528	0.0433	NS
14	0.7590	0.0467	NS
15	1	0.0500	NS

APPENDIX D

SIMULATION RESULTS FOR CHAPTER III

0% null

Power

	IND	BF	EF	B-H	ST
0.16	0.0268	0.0036	0.0039	0.0039	0.0042
0.37	0.1946	0.1039	0.109	0.1263	0.1346
0.501	0.3063	0.1453	0.1644	0.2039	0.2348
0.587	0.3671	0.1953	0.2192	0.2769	0.317

25% null

Power

	IND	BF	EF	B-H	ST
0.164	0.0586	0.011	0.0123	0.0121	0.0134
0.204	0.0677	0.0101	0.0118	0.0125	0.0144
0.362	0.276	0.147	0.1577	0.1857	0.2005
0.508	0.3866	0.1768	0.2059	0.2544	0.2965

Type I error

	IND	BF	EF	B-H	ST
0.164	0.0095	0.0008	0.0008	0.0008	0.001
0.204	0.01	0.0006	0.0008	0.0008	0.0011
0.362	0.01	0.0008	0.001	0.0024	0.0035
0.508	0.01	0.0013	0.0019	0.0042	0.0063

FDR

	IND	BF	EF	B-H	ST
0.164	0.0205	0.0023	0.0025	0.0023	0.0026
0.204	0.026	0.0022	0.0026	0.0025	0.0032
0.362	0.0098	0.001	0.0011	0.0023	0.003
0.508	0.0078	0.0019	0.0023	0.0038	0.005

50% null

Power

	IND	BF	EF	B-H	ST
0.084	0.0269	0.0028	0.0032	0.0032	0.0037
0.12	0.0532	0.0086	0.0092	0.0092	0.0098
0.24	0.243	0.0586	0.0688	0.078	0.0898
0.352	0.4173	0.2267	0.2472	0.2876	0.3091

Type I error

	IND	BF	EF	B-H	ST
0.084	0.01	0.001	0.0011	0.001	0.0011
0.12	0.0097	0.0008	0.0009	0.001	0.001
0.24	0.0098	0.0007	0.001	0.0015	0.0019
0.352	0.01	0.0008	0.0011	0.0025	0.0038

FDR

	IND	BF	EF	B-H	ST
0.084	0.0671	0.0078	0.0086	0.0078	0.0081
0.12	0.0544	0.0062	0.0067	0.0064	0.0069
0.24	0.0341	0.0047	0.0058	0.0061	0.0072
0.352	0.0192	0.0021	0.0027	0.0049	0.0072

75%null

Power

	IND	BF	EF	B-H	ST
0.04	0.0188	0.0013	0.0015	0.0015	0.0019
0.068	0.0911	0.0302	0.0313	0.0303	0.0313
0.128	0.2178	0.0611	0.065	0.0698	0.0751
0.264	0.8563	0.5791	0.6145	0.7023	0.7339

Type I error

	IND	BF	EF	B-H	ST
0.04	0.0095	0.0009	0.001	0.0009	0.001
0.068	0.01	0.0006	0.0008	0.0008	0.0009
0.128	0.01	0.0009	0.0011	0.0012	0.0015
0.264	0.0097	0.001	0.0013	0.0029	0.0038

FDR

	IND	BF	EF	B-H	ST
0.04	0.0984	0.0101	0.0115	0.01	0.0112
0.068	0.0928	0.0071	0.0079	0.0079	0.0085
0.128	0.0672	0.0091	0.0106	0.0103	0.0115
0.264	0.0274	0.0046	0.0051	0.0091	0.0114

Figure 39: Average power, type I error, and FDR with sample size 100 in the contingency table. First column of each table indicates the θ, defined in 4.4.1.

0% null

Power

	IND	BF	EF	B-H	ST
0.16	0.1395	0.0282	0.0332	0.0368	0.0456
0.37	0.4546	0.3101	0.3267	0.3967	0.4417
0.501	0.6606	0.5609	0.5755	0.6438	0.6802
0.587	0.7917	0.6756	0.7073	0.7835	0.8259

25% null

Power

	IND	BF	EF	B-H	ST
0.164	0.3912	0.1805	0.2121	0.2655	0.3047
0.204	0.4127	0.2089	0.2399	0.2967	0.3386
0.362	0.5654	0.4472	0.4572	0.5235	0.5554
0.508	0.9431	0.8271	0.8451	0.9323	0.9626

Type I error

	IND	BF	EF	B-H	ST
0.164	0.0086	0.0007	0.0013	0.0031	0.0051
0.204	0.01	0.0007	0.0015	0.0034	0.0062
0.362	0.01	0.0011	0.0015	0.0056	0.0107
0.508	0.0099	0.0008	0.0012	0.0077	0.015

FDR

	IND	BF	EF	B-H	ST
0.164	0.008	0.001	0.0013	0.0026	0.0034
0.204	0.006	0.001	0.0017	0.0028	0.0042
0.362	0.006	0.0007	0.001	0.0031	0.0053
0.508	0.0033	0.0003	0.0004	0.0026	0.0082

50% null

Power

	IND	BF	EF	B-H	ST
0.084	0.1687	0.0498	0.0554	0.0623	0.0686
0.12	0.3526	0.1215	0.139	0.1547	0.1888
0.24	0.8338	0.6413	0.6675	0.7748	0.8152
0.352	0.8481	0.7063	0.7219	0.8074	0.8374

Type I error

	IND	BF	EF	B-H	ST
0.084	0.01	0.0008	0.0011	0.0014	0.0017
0.12	0.0101	0.0005	0.0008	0.0015	0.0029
0.24	0.0091	0.0006	0.0008	0.0041	0.008
0.352	0.0094	0.0008	0.001	0.0045	0.0091

FDR

	IND	BF	EF	B-H	ST
0.084	0.0407	0.0047	0.0062	0.0058	0.0066
0.12	0.0227	0.0022	0.0032	0.0043	0.0068
0.24	0.0096	0.0008	0.0011	0.0046	0.0083
0.352	0.0097	0.001	0.0012	0.0047	0.0091

75%null

Power

	IND	BF	EF	B-H	ST
0.04	0.0637	0.0098	0.0113	0.0121	0.0134
0.068	0.7391	0.3472	0.3953	0.463	0.504
0.128	0.8781	0.6746	0.6918	0.7645	0.7853
0.264	1	0.999	0.999	0.999	0.999

Type I error

	IND	BF	EF	B-H	ST
0.04	0.01	0.0006	0.0007	0.0007	0.0008
0.068	0.0099	0.0007	0.0011	0.0021	0.0033
0.128	0.0093	0.0006	0.0008	0.0022	0.0035
0.264	0.0095	0.0006	0.0007	0.0032	0.005

FDR

	IND	BF	EF	B-H	ST
0.04	0.0924	0.0061	0.0071	0.0065	0.0075
0.068	0.0297	0.004	0.0055	0.0076	0.01
0.128	0.0249	0.0023	0.0028	0.006	0.0088
0.264	0.0218	0.0015	0.0018	0.0076	0.0114

Figure 40: Average power, type I error, and FDR with sample size 500 in the contingency table. First column of each table indicates the θ, defined in 4.4.1.

0% null

Power

	IND	BF	EF	B-H	ST
0.16	0.3171	0.1095	0.1394	0.1786	0.2247
0.37	0.6296	0.4714	0.4969	0.5967	0.6514
0.501	0.7349	0.6706	0.674	0.7258	0.7595
0.587	0.8551	0.8055	0.808	0.8517	0.8855

25% null

Power

	IND	BF	EF	B-H	ST
0.164	0.5914	0.4509	0.4732	0.5491	0.5804
0.204	0.6001	0.4685	0.4891	0.5627	0.5904
0.362	0.6887	0.5901	0.5994	0.6656	0.6911
0.508	0.9976	0.9817	0.9833	0.9968	0.9992

Type I error

	IND	BF	EF	B-H	ST
0.164	0.0093	0.0004	0.0006	0.0038	0.0094
0.204	0.0098	0.0006	0.001	0.0042	0.0095
0.362	0.01	0.0008	0.0012	0.0068	0.0144
0.508	0.0098	0.0006	0.0008	0.0076	0.0319

FDR

	IND	BF	EF	B-H	ST
0.164	0.0047	0.0003	0.0004	0.002	0.0046
0.204	0.0049	0.0004	0.0006	0.0022	0.0045
0.362	0.005	0.0004	0.0006	0.0029	0.0059
0.508	0.003	0.0002	0.0003	0.0024	0.0097

50% null

Power

	IND	BF	EF	B-H	ST
0.084	0.3296	0.1686	0.1811	0.206	0.2205
0.12	0.6757	0.3894	0.4349	0.5372	0.5944
0.24	0.9653	0.8913	0.8968	0.9503	0.9643
0.352	0.9643	0.8864	0.8914	0.948	0.9634

Type I error

	IND	BF	EF	B-H	ST
0.084	0.01	0.0004	0.0006	0.0014	0.0021
0.12	0.01	0.0008	0.0014	0.0038	0.0069
0.24	0.01	0.0008	0.001	0.0054	0.0112
0.352	0.0096	0.0007	0.001	0.0048	0.0095

FDR

	IND	BF	EF	B-H	ST
0.084	0.0275	0.0013	0.0021	0.0034	0.0047
0.12	0.0132	0.0017	0.0028	0.0054	0.0085
0.24	0.0094	0.0008	0.001	0.005	0.01
0.352	0.0088	0.0007	0.001	0.0048	0.0095

75%null

Power

	IND	BF	EF	B-H	ST
0.04	0.1358	0.0262	0.0294	0.0316	0.0353
0.068	0.9495	0.8448	0.8532	0.9036	0.9142
0.128	0.9891	0.945	0.947	0.9718	0.9763
0.264	1	1	1	1	1

Type I error

	IND	BF	EF	B-H	ST
0.04	0.0096	0.0006	0.0006	0.0007	0.0008
0.068	0.0098	0.0007	0.0008	0.0027	0.0041
0.128	0.0095	0.0007	0.0008	0.0028	0.0043
0.264	0.01	0.0006	0.0007	0.0037	0.0051

FDR

	IND	BF	EF	B-H	ST
0.04	0.0767	0.006	0.0067	0.0062	0.0072
0.068	0.0239	0.0019	0.0023	0.0067	0.01
0.128	0.0222	0.0018	0.0021	0.0067	0.01
0.264	0.0231	0.0014	0.0016	0.0086	0.01

Figure 41: Average power, type I error, and FDR with sample size 1000 in the contingency table. First column of each table indicates the θ, defined in 4.4.1.

0.9%null

Power					
	IND	BF	EF	B-H	ST
1	0.0608	0.0008	0.0008	0.0008	0.0008
2	0.2842	0.0114	0.0148	0.0148	0.015
3	0.6582	0.1008	0.1984	0.2362	0.2422
4	0.9198	0.3922	0.6132	0.7316	0.7418

Type I error					
	IND	BF	EF	B-H	ST
1	0.01	0	0	0	0.0001
2	0.01	0	0	0	0
3	0.0091	0	0.0001	0.0003	0.0003
4	0.01	0.0001	0.0002	0.0008	0.0009

FDR					
	IND	BF	EF	B-H	ST
1	0.6184	0.02	0.02	0.02	0.03
2	0.249	0.01	0.01	0.01	0.01
3	0.1106	0.0034	0.0048	0.0078	0.0082
4	0.0875	0.0019	0.0022	0.0091	0.01

0.8%null

Power					
	IND	BF	EF	B-H	ST
1	0.0561	0.0006	0.0006	0.0006	0.0006
2	0.278	0.0136	0.0294	0.0289	0.0309
3	0.6661	0.1036	0.2903	0.3546	0.3741
4	0.9212	0.3978	0.6807	0.8037	0.8233

Type I error					
	IND	BF	EF	B-H	ST
1	0.0097	0	0	0	0
2	0.0099	0	0	0.0001	0.0001
3	0.0099	0	0.0002	0.0008	0.001
4	0.01	0.0001	0.0003	0.0017	0.0024

FDR					
	IND	BF	EF	B-H	ST
1	0.4075	0.01	0.01	0.01	0.01
2	0.1255	0.01	0	0.0033	0.0033
3	0.0559	0.0005	0.0027	0.0084	0.0095
4	0.0447	0.0005	0.0019	0.0081	0.0111

0.6%null

Power					
	IND	BF	EF	B-H	ST
1	0.0595	0.0004	0.0006	0.0004	0.0004
2	0.287	0.0111	0.0436	0.0402	0.0454
3	0.66	0.1022	0.3645	0.4521	0.5027
4	0.9206	0.395	0.7308	0.8576	0.8901

Type I error					
	IND	BF	EF	B-H	ST
1	0.0095	0	0	0	0
2	0.0095	0	0.0001	0.0002	0.0003
3	0.01	0	0.0003	0.0014	0.0023
4	0.01	0	0.0004	0.0033	0.0054

FDR					
	IND	BF	EF	B-H	ST
1	0.1925	0.01	0.01	0.01	0.01
2	0.047	0.0033	0.0041	0.0064	0.0082
3	0.0221	0	0.0011	0.0046	0.0068
4	0.0165	0	0.0008	0.0058	0.009

Figure 42: Average power, type I error, and FDR with sample size from normal random variable. First column of each table indicates the δ, defined in 4.4.3.

0.4%null

Power

	IND	BF	EF	B-H	ST
1	0.0583	0.0008	0.0009	0.0009	0.0009
2	0.2836	0.0119	0.0636	0.0571	0.068
3	0.6648	0.1009	0.4195	0.519	0.603
4	0.9234	0.3943	0.7664	0.889	0.9343

Type I error

	IND	BF	EF	B-H	ST
1	0.01	0	0	0	0
2	0.0097	0	0.0001	0.0005	0.0006
3	0.01	0	0.0004	0.0033	0.0059
4	0.01	0	0.0004	0.006	0.0138

FDR

	IND	BF	EF	B-H	ST
1	0.113	0	0	0	0
2	0.0224	0	0.0019	0.0058	0.0067
3	0.0101	0	0.0007	0.0041	0.0064
4	0.0075	0	0.0003	0.0044	0.0097

0.2%null

Power

	IND	BF	EF	B-H	ST
1	0.0583	0.0005	0.0004	0.0005	0.0005
2	0.2759	0.012	0.0789	0.0775	0.0979
3	0.6641	0.1004	0.4535	0.5611	0.6818
4	0.9238	0.3945	0.7865	0.9072	0.966

Type I error

	IND	BF	EF	B-H	ST
1	0.0095	0	0	0	0
2	0.01	0	0.0005	0.0009	0.0015
3	0.0088	0	0.0008	0.0028	0.0105
4	0.01	0	0.001	0.0071	0.0274

FDR

	IND	BF	EF	B-H	ST
1	0.0371	0	0	0	0
2	0.0107	0	0.0014	0.0024	0.0033
3	0.0033	0	0.0004	0.0013	0.0038
4	0.0027	0	0.0003	0.002	0.007

0.002%null

Power

	IND	BF	EF	B-H	ST
1	0.0564	0.0007	0.0011	0.0009	0.001
2	0.2823	0.0125	0.0991	0.0999	0.1329
3	0.6645	0.1016	0.4824	0.5987	0.7629
4	0.9225	0.3971	0.8023	0.9183	0.9934

Type I error

	IND	BF	EF	B-H	ST
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0.03	0	0	0.03	0.19

FDR

	IND	BF	EF	B-H	ST
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0.0001	0	0	0.0001	0.0004

Figure 43: Average power, type I error, and FDR with sample size from normal random variable. First column of each table indicates the δ, defined in 4.4.3.

APPENDIX E

DESCRIPTION OF DSSP FOR CHAPTER III

The DSSP is a program, which defines secondary structure, geometrical features and solvent exposure of proteins, given atomic coordinates in Protein Data Bank format. Figure 44 shows an output file of DSSP. Followings are a brief description of the output. For more details, see Kabsch and Sander (1983).

1. *# RESIDUE:* It involves two columns. First column is sequential amino acid number including chain breaks. Second column indicates an original PDB residue sequence number, which is not necessarily sequential.

2. *# AA:* One letter amino acid code.

3. *# STRUCTURE:* It contains eight columns containing types of secondary structure, 3-turns helix, 4-turns helix, 5-turns-helix, geometrical bend, chirality, and beta bridge labels (last two columns). The labels give two types of alphabet: small letter (a, b, c, ...) for parallel strands and capital letter (A, B, C, ...) for antiparallel strands.

4. *BP1 and BP2:* These give the amino acid number of first and second bridge partner followed by one letter sheet label.

5. *Other columns:* Other columns present geometrical features of the amino acids.

```
#  RESIDUE AA STRUCTURE BP1 BP2  ACC     N-H-->O    O-->H-N    N-H-->O    O-->H-N    TCO  KAPPA ALPHA  PHI   PSI    X-CA   Y-CA   Z-CA
  1    1 A M              0   0   74      0, 0.0    20,-0.3     0, 0.0     3, 0.0   0.000 360.0 360.0 360.0 150.9   19.0 30.9   33.4
  2    2 A N     > -      0   0   59      1,-0.1     4,-2.6   156,-0.1     5,-0.2  -0.919 360.0 -73.7-164.9 178.5   21.6   32.8   35.3
  3    3 A I   H > S+     0   0   23     -2,-0.3     4,-2.8     1,-0.2     5,-0.3   0.880 127.1  53.5 -56.3 -38.3   25.3   32.9   36.2
  4    4 A F   H > S+     0   0   64      1,-0.2     4,-2.1     2,-0.2    -1,-0.2   0.948 114.0  39.0 -62.3 -46.2   24.7   29.9   38.6
  5    5 A E   H > S+     0   0   94      2,-0.2     4,-1.6     1,-0.2    -1,-0.2   0.888 115.4  54.5 -69.8 -34.9   23.1   27.8   36.0
  6    6 A M   H X S+     0   0    0     -4,-2.6     4,-2.0     2,-0.2    -2,-0.2   0.906 110.1  45.5 -64.1 -47.5   25.5   29.0   33.3
  7    7 A L   H X>S+     0   0    0     -4,-2.8     4,-3.4     2,-0.2     5,-0.7   0.872 108.3  55.6 -65.3 -43.1   28.6   28.1   35.4
  8    8 A R   H X5S+     0   0   94     -4,-2.1     4,-1.6    -5,-0.3    -1,-0.2   0.896 110.5  48.1 -56.2 -38.9   27.3   24.6   36.4
  9    9 A I   H <5S+     0   0   74     -4,-1.6    -2,-0.2    -5,-0.2    -1,-0.2   0.960 116.6  41.5 -66.2 -50.4   26.9   24.0   32.7
 10   10 A D   H <5S+     0   0    2     -4,-2.0    -2,-0.2     1,-0.2    -1,-0.2   0.823 129.2  26.6 -67.0 -39.1   30.3   25.2   31.8
 11   11 A E   H <5S-     0   0   22     -4,-3.4    19,-0.3     1,-0.1    -3,-0.2   0.673  95.5-152.1 -99.9 -21.5   32.3   23.6   34.7
 12   12 A G     <<-      0   0   13     -4,-1.6     2,-0.5    -5,-0.7    -1,-0.1  -0.149  25.5 -80.7  75.4-178.6   30.0   20.8   35.4
 13   13 A L       +      0   0   68    198,-0.2     2,-0.4    16,-0.1    16,-0.2  -0.927  46.7 173.5-130.6 112.7   29.7   19.3   38.9
 14   14 A R   E    -A    28   0A  20     14,-2.3    14,-2.0    -2,-0.5     4,-0.1  -0.933  20.2-164.9-118.7 136.9   32.3   16.8   40.3
 15   15 A L   E    S+     0   0   61     -2,-0.4    43,-1.3    12,-0.1     2,-0.3   0.482  75.5  55.0 -96.5  -1.3   32.4   15.4   43.8
 16   16 A K   E    S-C   57   0B 133     41,-0.2    41,-0.2    12,-0.1    12,-0.2  -0.916 102.4 -78.3-127.9 155.6   35.9   14.0   43.7
 17   17 A I   E     +     0   0   15     39,-1.0     2,-0.3    -2,-0.3    10,-0.2  -0.186  57.8 168.4 -52.0 137.9   39.2   15.7   42.8
 18   18 A Y   E    -A    26   0A   4      8,-2.4     8,-3.4    -4,-0.1     2,-0.4  -0.913  35.0-102.4-145.7 169.8   39.8   16.2   39.0
 19   19 A K   E    -A    25   0A  97     -2,-0.3     6,-0.2     6,-0.2     2, 0.0  -0.854  31.1-144.8-102.1 138.3   42.0   18.1   36.6
 20   20 A D    >   -      0   0   35      4,-1.7     3,-0.9    -2,-0.4   187,-0.1   0.078  44.5 -76.1 -82.0-168.9   40.8   21.2   34.9
 21   21 A T   T 3  S+     0   0   20      1,-0.2   183,-0.3     2,-0.1   182,-0.2   0.633 135.0  48.2 -71.1 -14.7   41.7   22.4   31.4
 22   22 A E   T 3  S-     0   0   91    181,-0.2    -1,-0.2     2,-0.1   182,-0.1   0.414 121.4-105.9 -99.7  -9.1   45.1   23.4   32.5
 23   23 A G   S <  S+     0   0   14     -3,-0.9     2,-0.3     1,-0.3    -2,-0.1   0.498  73.3 141.6  93.4   7.5   45.8   20.1   34.3
 24   24 A Y       -      0   0   81      1,-0.1    -4,-1.7    -5, 0.0    -1,-0.3  -0.623  60.7 -96.9 -87.6 145.2   45.3   21.6   37.9
 25   25 A Y   E     +AB  19   34A  33      9,-0.6     8,-3.8    11,-0.4     9,-1.1  -0.337  55.8 163.0 -61.4 118.9   43.5   19.9   40.8
 26   26 A T   E    -AB  18   32A   2     -8,-3.4    -8,-2.4     6,-0.3     2,-0.3  -0.834  17.3-167.1-132.6 167.6   39.9   21.1   41.1
 27   27 A I   E  > - B   0   31A   0      4,-1.3     4,-2.7    -2,-0.3     2,-0.2  -0.986  51.3  -0.7-156.4 164.2   36.7   20.0   42.7
 28   28 A G   E  4 S-A   14   0A   1    -14,-2.0   -14,-2.3    -2,-0.3     2,-0.9  -0.407 127.4  -7.3  61.4-126.4   32.9   20.6   42.8
 29   29 A I   T  4 S-     0   0    2     34,-0.3    -1,-0.2   -16,-0.2   -16,-0.1  -0.698 129.1 -54.0-103.5  71.9   32.0   23.3   40.2
 30   30 A G   T  4 S+     0   0   15     -2,-0.9     2,-0.9   -19,-0.3    -2,-0.2   0.770  79.9 161.8  65.0  36.2   35.4   24.4   39.2
 31   31 A H   E  < -B   27   0A  34     -4,-2.7    -4,-1.3     1, 0.0    -1,-0.2  -0.757  35.7-137.0 -86.6 105.6   36.9   25.2   42.6
 32   32 A L   E    -B   26   0A  68     -2,-0.9    -6,-0.3    -6,-0.2     3,-0.1  -0.382  21.8-175.5 -61.6 135.6   40.6   25.3   42.1
 33   33 A L   E    -     0   0   16     -8,-3.8     2,-0.3     1,-0.4    -7,-0.2   0.846  57.7 -34.5 -97.4 -46.5   42.5   23.5   44.8
 34   34 A T   E    -B   25   0A  27     -9,-1.1    -9,-0.6     2,-0.1    -1,-0.4  -0.984  34.0-130.6-168.4 163.8   46.1   24.1   43.9
```

Figure 44: Description of DSSP.

REFERENCES

[1] Agresti, A. An Introduction to Categorical Data Analysis. John Wiley & Sons Inc., New York, 1996.

[2] Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M., Robert, K., and Walter, P. Essential Cell Biology: An Introduction to the Molecular Biology of the Cell. Garland Publishing Inc., New York, 1997.

[3] Baldi, P., Pollastri, G., Andersen, C.A.F., and Brunak, S. Matching protein β-Sheet partners by feedforward and recurrent neural networks. *ISBM*, 25–36, 2000.

[4] Benjamini, Y., and Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of Royal Statistical Society, Series B*, 57, 289–300, 1995.

[5] Benjamini, Y., and Yekutieli, D. The control of the false discovery ratein multiple testing under dependency. *Annals of Statistics*, 29, 1165–1188, 2001.

[6] Berry, M.J.A., and Linoff, G.S. *Mastering Data Mining.* Wiley, New York, 2000.

[7] Bishop, Y.V.V., Fienberg, S.E., and Holland, P.W. Discrete Multivarite Analysis. MIT Press, Cambridge, MA, 1975.

[8] Blader, M., Zhang, X.J., and Matthews, B.W. Structural basis of amino acid αhelix propensity. *Science*, 260, 1637–1640, 1993.

[9] Blake, C.L., and Merz, C.J. UCI Repository of machine learning databases. Irvine, CA. http://www.ics.uci.edu/ ∼mlearn/MLRepository.html. 1998.

[10] Breiman, L., Friedman, J., Olshen, R., and Stone C. Classification and Regression Trees. Wadsworth International Group, Belmont, CA, 1984.

[11] Breiman, L., and Spector, P. Submodel selection and evaluation in regression: the X-random case. *Intern. Stat. Rev.*, 60, 291–319, 1992.

[12] Brown, T.A. Genomes. Wiley-Liss, Oxford, UK, 2002.

[13] Buja, A., and Lee, Y.S. Data mining criteria for tree-based regression and classification. *Proceedings of KDD-2001*, 27-36, 2001.

[14] Casella G., and Berger, R.L. Statistical Inference. Duxbury, Pacific Grove, CA, 2002.

[15] Cestnik, B., and Bratko, I. On estimating probabilities in tree pruning. *Proceedings of Fifth European Working Session in Learning*, Springer-Verlag, 1991.

[16] Clark, L.A., and Pregibon, D. Statistcal Models in S. Chapter 9: Tree-Based Models. Wadsworth and Brooks, Pacific Grove, CA, 1992.

[17] Cochran, W. G. Some methods of strengthening the common χ^2 tests. *Biometrics*, 10, 417–451, 1954.

[18] Coifman, R. R., and Wickerhauser, M Entropy-based algorithms for best basis selection. *IEEE Transaction on Information Theory*, 38, 713–718, 1992.

[19] Crawford S. Extensions to the CART algorithm. *International Journal of Man-Machine Studies*, 31, 197–217, 1989.

[20] Creamer, T.P., and Rose, G.D. α-helix forming propensities in peptides and proteins. *Proteins*, 19, 85–97, 1994.

[21] Donoho, D. L. CART and best-ortho-basis: a connection. *Annals of Statistics,* 25, 1870–1911, 1997.

[22] Donoho, D. L. Wedgelets: nearly minimax estimation of edges. *Annals of Statistics,* 27, 859–897, 1999.

[23] Dudoit, S., Shaffer, J.P., and Boldrick, J.C. Multiple Hypothesis Testing in Microarray Experiments. *Statistical Science,* 18, 71–103, 2003.

[24] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. Biological sequence analysis. Cambridge University Press, United Kingdom, 1998.

[25] Efron, B. Estimating the error rate of a prediction rule: Improvement of cross-validation. *Journal of the American Statistical Association*, 78, 316–331, 1983.

[26] Efron, B. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81, 461–470, 1986.

[27] Efron, B. Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association*, 99, 99–104, 2004.

[28] Efron, B., and Tibshirani R. Microarrays, empirical Bayes methods, and false discovery rates. Technical Report 2001-217, Department of Statistics, Stanford University, 2001.

[29] Efron, B., Tibshirani, R., Storey, J.D., and Tusher, V. Empirical Bayes Analysis of a Microarray Experiment. *Journal of American Statistical Association*, 96, 1151–1160, 2001.

[30] Frishman, D., and Argos, P. Incorporation of non-local interactions in protein secondary structure prediction. *Protein Engineering*, 9, 133–142, 1996.

[31] Genovese, C. R., Lazar, N. A., and Nichols, T. Thresholding of statistical maps in functional neuroimaging using the false discovery rate. *Neuroimaging*, 15, 870–878, 2002.

[32] Haberman, S. J. The analysis of residuals in cross-classified tables. *Biometrics*, 29, 205–220, 1973.

[33] Hamilton, H.J. C4.5. http://www2.cs.uregina.ca/∼hamilton/courses/831/notes/ml/dtrees/c4.5/tutorial.html, 2004.

[34] Hastie, T., Tibshirani, R., and Friedman, J. H. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer series in statistics. New York, 2001.

[35] Huo, X., Kim, S.B., Tsui, K. L., and Wang, S. FBP: A frontier-based tree-prunung algorithm. *INFORMS Journal on Computing*, accepted, 2004.

[36] Hutchinson E.G., Sessions, R.B., Thornton, J.M., and Woolfson, D. N. Determinants of strand register in antiparallel β-sheets of proteins. *Protein Science*, 7, 2287–2300, 1998.

[37] Jonathan, P., Krzanowski, W.J., and McCarthy W.V. On the use of cross-validation to assess performance in multivariate prediction. *Statistics and Computing*, 10, 209–229, 2000.

[38] Kabsh, W., and Sander, C. Dictionary of proteins secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22, 2577–2637, 1983.

[39] Kass, G. V. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29, 119–127, 1980.

[40] Kim, H., and Koehler, G. J. An investigation on the conditions of pruning an induced decision tree. *European Journal of Operational Research*, 77, 82–95, 1994.

[41] Lachenbruch, P.A., and Mickey, M.R. Estimation of Error Rates in Discriminant Analysis. *Technometrics*, 10, 1–11, 1968.

[42] Lancaster, M.B. The derivation and partition of χ^2 in certain discrete distributions. *Biometrika*, 36, 117–129, 1949.

[43] Li, X. B., Sweigart, J., Teng, J., Donohue, J., and Thombs, L. A dynamic programming based pruning method for decision trees. *INFORMS Journal on Computing*, 13, 332-344, 2001.

[44] Lifson, S., and Sander, C. Specific recognition in the tertiary structure of β-sheets of proteins. *Journal of Molecular Biology*, 139, 627–639, 1980.

[45] Lim, T.-S., Loh, W.Y., and Shih, Y.S. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40, 203-229, 2000.

[46] Loh, W.-Y., and Vanichsetakul, N. Tree-structured classification via generalied discriminant analysis. *Journal of the American Statistical Association*, 83, 715–728, 1988.

[47] Luenberger, D. G. Investment Theory. Oxford University Press. New York, 1998.

[48] Markowitz, H. Portfolio Selection *Journal of Finance*, 7, 77-91, 1952.

[49] Mehta, M., Rissanen, J., and Agrawal, R. MDL-based decision tree pruning. *Proceedings of KDD-1995*, 216-221, 1995.

[50] Minger, J. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3, 319-342, 1989a.

[51] Minger, J. An Empirical comparison of pruning methods for decision-tree induction. *Machine Learning*, 4, 227-243, 1989b.

[52] Minor, D.L. Jr., and Kim, P.S. Measurement of the β-sheet forming propensities of amino acids. *Nature*, 367, 60–663, 1994.

[53] Mitchell, T. M. *Machine Learning.* McGraw-Hill, New York, 1997.

[54] Mount, D. W. Bioinformatics: Sequence and Genome Analysis. Gold Spring Harbor Laboratory Press, New York, 2001.

[55] Quinlan, J. R. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27, 221-234, 1987.

[56] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1992.

[57] Racine, J. Consistent cross-validatory model-selection for dependent data: hv-block cross-validaion. *Journal of Econometrics*, 99, 39–61, 2000.

[58] Rastogi, R., and Shim, K.S. PUBLIC: A decision tree classifier that integrates building and pruning. *VLDB*, 404-415, 1998.

[59] Salford Systems. CART and MARS. http://www.salford-systems.com/. 2004.

[60] Schmidler, S.C., Liu, J.S., and Brutlag, D.L. Baysian Segmentation of Protein Secondary Structure. *Jounral of Computational Biology*, 7, 233–248, 2000.

[61] Sethi, I.K., and Sarvarayudu, G.P.R. Hierarchical classifier design using mutual information. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 4, 441–445, 1982.

[62] Shaffer, J. Multiple hypothesis testing. *Annu. Rev. Psychol.*, 46, 561–584, 1995.

[63] Shao, J. Linear-model selection by cross-validation. *Journal of the American Statistical Association*, 88, 486–494, 1993.

[64] Shao, J. Bootstrap model selection. *Journal of the American Statistical Association*, 91, 655–665, 1996.

[65] Shao, J. Convergence rates of the generalization information criterion. *Journal of Nonparametric Statistics*, 9, 217–22, 1998.

[66] Smith, C.K., and Regan, L. Guidelines for protein design - The energetics of beta-sheet side-chain interactions. *Science*, 270, 980–982, 1995.

[67] Stone, M. Cross-validatory choice and assessment of statistical prediction. *Journal of the Royal Statistical Society*, Series B, 36, 111–133, 1974.

[68] Storey, J.D. A direct approach to false discovery rates. *Journal of Royal Statistical Society, Series B*, 64, 479–498, 2002.

[69] Storey, J.D. The positive false discovery rate: a Baysian interpretation and the q-value. *The Annals of Statistics*, 31, 2013–2035, 2003.

[70] Storey, J.D., and Tibshirani, R. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100, 9440–9445, 2003.

[71] Storey, J.D., Taylor, J.E., and Siegmund, D. Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unifed approach. *Journal of Royal Statistical Society, Series B,* 66, 187–205, 2004.

[72] Tusher, V.G., Tibshirani, R., and Chu, G. Significance analysis of microarray applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, 98, 5116–5121, 2001.

[73] Vehtari, A., and Lampinen, J. Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, 14, 2439–2468, 2002.

[74] von Heijne, G., and Blomberg, C. The β-structure: Inter-strand Correlations. *Journal of Molecular Biology*, 117, 821–824, 1977.

[75] von Heijne, G., and Blomberg, C. Some Global β-sheet Characteristics. *Bipolymers*, 17, 2033–2037, 1978.

[76] Wink, A.M., and Roerdink, Jos B. T. M. Denoising functional MR images: a comparison wavelet denoising and gaussian smooting. *IEEE Transactions on Medical Imaging*, 23, 374–387, 2004.

[77] Wolberg, W. H., and Mangasarian, O. L. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87, 9193-9196, 1990.

[78] Wouters, M.A., and Curmi, P.M.G. An analysis of side-chain interactions and pair correlations within antiparallel beta-sheets: The differences between backbone hydrogen bonded and non-hydrogen-bonded residue pairs. *Proteins*, 22, 119–131, 1995.

[79] Yekutieli, D., abd Benjamini, Y. Resampling-based false discovery rate controlling multiple test procedures for correlated test statistics. *Journal of Statistical Planning and Inference*, 82, 171–196, 1999.

[80] Zhang, P. On the distributional properties of model selection criteria. *Journal of the American Statistical Association*, 87, 732–737, 1992.

[81] Zhang, P. Model selection via multifold cross validation. *The Annals of Statistics*, 21, 299–313, 1993a.

[82] Zhang, P. On the convergence rate of model selection criteria. *Commun. Statist. – Theory Meth.*, 22, 2765–2775, 1993b.

[83] Zhou, H., and Braun, W. Sequence specificity, statistical potentials, and three-dimensional structure prediction with self-correcting distance geometry calculations of β-sheet formation in proteins. *Protein Science*, 8, 326–342, 1999.

[84] Zhou, X., Alber, F., Folkers, G., Gonnet, G., and Chelvanayagam, G. An analysis of the helix-to-strand transition between peptides with identical sequence. *Proteins*, 41, 248–256, 2000.

VITA

Seoung Bum Kim received his B.S. in Industrial Engineering from Hanyang University in 1999. In 2001 and 2004, he completed an M.S. in the Industrial and Systems Engineering and Statistics, respectively at the Georgia Institute of Technology, where he has worked as a graduate research assistant pursuing a Ph.D. degree concentrated on Engineering Statistics since the summer of 2001. His main research focus is computational investigations of data mining algorithms and multiple testing in large-scale problems. Main applications include biomedical, health, manufacturing, and service systems.